

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

УДК: 004.42

«До захисту допущено»
В.о. завідувача кафедри
О.А.Павлов
(підпис) (ініціали, прізвище)
“ ” 2019 р.

Дипломний проект
на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 «Комп'ютерні науки»

на тему: «Інформаційна система підтримки процесу подачі оголошень»

Виконав:

студент 4 курсу, групи ІС-52

Уманський Володимир Андрійович
(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Цюцюра М.І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н., доц. Тєлишева Т.О.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент

с.н.с., к.т.н. Кондратова Л.П.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент Уманський В. А.

(підпис)

Київ – 2019 р.

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 125 сторінок, 32 рисунка, 29 таблиць, 2 додатки, 13 джерел.

Дипломний проект присвячений розробці інформаційної системи керування процесу подачі оголошень.

В дипломному проекті було розглянуто : алгоритми та методи колаборативної фільтрації, усі можливі методи сортування, було розглянуто підходи до розробки подібної системи, та реалізовано як і дружній інтерфейс користувача, так і функціональну модель системи.

У розділі з інформаційного забезпечення були визначені вхідні та вихідні дані до комплексу задач, було розроблено базу даних, що складається з 19 таблиць з використанням мови MySQL для збереження даних, яка відповідає поставленим цілям проекту.

У розділі з математичного забезпечення було обґрунтовано створений метод колаборативної фільтрації. Було проведене порівняння з існуючими методами, та об'єктивно обрано найкращий.

У розділі з програмного забезпечення описані основні засоби розробки комплексу задач, висунуті вимоги до технічного забезпечення, обрано та обґрунтовано архітектуру програмного забезпечення.

У технологічному розділі описана інструкція користувача та адміністратора і проведене тестування системи.

ДОШКА ОГОЛОШЕНЬ, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, РЕКОМАНДАЦІЙНА СИСТЕМА, ЦІЛЬОВА ПРОПОЗИЦІЯ

					ДП ІС-5223.1181-с.ПЗ				
		Прізвище	Підпис	Дата					
Розроб.	Уманський В.А.				Інформаційна система підтримки процесу подачі оголошень	Літ.		Арк.	Аркушів
Перевірив.	Цюцюра М.І.							2	91
Н. кон.	Халус О.А.					КПІ ім.І.Сікорського ФІОТ кафедра АСОІУ гр. ІС-52			
Затв.	Павлов О.А.								

ABSTRACT

Structure and scope of work. The explanatory note of the diploma project consists of five sections, containing 125 pages, 32 figures, 29 tables, 2 annexes, 13 sources.

The thesis project is devoted to the development of an information management system for the submission of advertisements. The graduation project reviewed algorithms and methods of collaborative filtration, all possible sorting methods, considered approaches to the development of such a system, and realized both the user interface and the functional model of the

system.

In the information support section, the input and output data were identified in a set of tasks, a database of 19 tables was created with the use of MySQL to store data that meets the objectives of the project.

In the section on mathematical support, a collaborative filtering method was substantiated. A comparison was made with existing methods, and the best was objectively chosen.

The software section describes the main tools for developing a set of tasks, the requirements for technical support, and chooses and justifies the software architecture.

The technology section —describes the user and administrator instructions and tests the system.

BASKET ADVERTISING, COLLABORATE FILTRATION,
RECOMMENDATION SYSTEM, TARGET OFFER

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП.....	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	7
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА.....	7
1.3.1 Призначення розробки.....	13
1.3.2 Цілі та задачі розробки.....	13
Висновок до розділу	13
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	14
2.1 ВХІДНІ ДАНІ.....	14
2.2 ВИХІДНІ ДАНІ.....	14
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ.....	15
Висновок до розділу	20
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	21
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	21
3.2 АЛГОРИТМИ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ.....	23
3.2 МАТЕМАТИЧНА МОДЕЛЬ.....	25
3.3 ОГЛЯД МЕТОДІВ РОЗВ'ЯЗАННЯ.....	27
3.4 АЛГОРИТМ ЗАСТОСУВАННЯ МОДИФІКОВАНОГО ПІДХОДУ.....	29
3.5 РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ ЕФЕКТИВНОСТІ МЕТОДУ.....	30
Висновок до розділу 34	
4. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	35
4.1 ЗАСОБИ РОЗРОБКИ.....	35
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ.....	44
4.3 ДІАГРАМА КОМПОНЕНТІВ	45
4.3.1 Особливості використання діаграми послідовності.....	45
4.3.2 Особливості використання діаграми класів.....	45
4.3.3 Особливості використання діаграми компонентів.....	46
4.4 Діаграма класів.....	46
4.5 Діаграма компонентів.....	47
4.6 Специфікація функцій.....	48
Висновок до розділу	52

5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	53
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА.....	53
5.2	<i>Мета випробувань</i>	<i>64</i>
5.3	<i>Загальні положення</i>	<i>64</i>
5.4	<i>Результати випробувань</i>	<i>64</i>
	ВИСНОВОК ДО РОЗДІЛУ	73
	ЗАГАЛЬНІ ВИСНОВКИ	74
	ПЕРЕЛІК ПОСИЛАНЬ	75
	ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ	76

ВСТУП

Оголошення – невеликі за обсягом текстові записи, які містять різноманітну інформацію, зазвичай з приводу продажу товарів, пропозиції послуг, пошуку роботи. У минулому сторіччі вони подавались в газетах, але з розвитком технологій почали з'являтися на теренах Інтернету. Мета оголошень дуже проста - проінформувати зацікавлене коло осіб про різні події, наприклад : навчання, роботу, продаж, купівлю чи оренду житла, про цікаві наукові заходи чи культурні події. Будь – яке оголошення – це коротке повідомлення (пропозиція, або інакша інформація), яка подається разом із контактними даними в залежності від типу оголошення [1].

Дипломний проект присвячений розробці системи керування процесом подачі оголошень з рекомендаційною системою.

Цілі створення Для досягнення цілей у роботі вирішуються наступні задачі: колаборативна фільтрація, керування доступом , пошук подібних запитів користувачів, їхнє порівняння та прогнозування вибору іншого споживача.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

1.1 Опис предметного середовища

У всі часи люди бажали налагодити торгівельні зв'язки одне з одним, тому що так раніше жив світ і сьогодні мало що змінилось. У всі часи існували люди, які бажають продати товари, послуги, а також споживачі цих товарів та послуг. Зараз все це також має місце бути, але змінилась форма взаємодії продавців і покупців. Сьогодні - це сайти-дошки оголошень в мережі .

Онлайн дошка оголошень - це веб-сайт, на якому публікуються юридичними або фізичними особами оголошення, поділені за тематичними розділами. Контент дошок оголошень може мати комерційний або некомерційний характер, а подача самих оголошень - платною або безкоштовною.

Принцип роботи онлайн дошок оголошень простий і єдиний для всіх: користувач вводить в підготовленої сайтом формі свої дані - ім'я або назву установи, якщо це комерційне оголошення, пошту, або контактний телефон, тему та опис товару , пропозиції або ж послуги, що користувач готовий надати. На головній сторінці відображаються зазвичай тільки тема і зображення. Щоб переглянути всі оголошення, необхідно пройти по посиланню, що веде до нього. Далі, користувач отримує розгорнуту інформацію про продавця і товар.

Різновиди дошок оголошень в мережі:

Безкоштовні дошки оголошень в даний момент користуються великою популярністю, як у комерційних так і ні цілях, у звичайних користувачів. На таких дошках будь-хто може безкоштовно розмістити своє оголошення, тільки зареєструвавшись. Можна також скористатись платними сервісами, щоб досягти більшої кількості переглядів свого оголошення.

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Платні дошки оголошень не дають можливості безкоштовно пропонувати будь які товари та послуги, тому не користуються наскільки великою популярністю, як безкоштовні.

Дошки, що не модеруються – дошки, на яких розміщується все, і ніхто не слідкує за політикою розміщення оголошень, ні за їхньою кількістю. Тому вони швидко стають звалищем непотрібної нікому інформації, тому не є наскільки потрібними в сьогоdnішньому суспільстві.

Дошки, що модеруються – більш зручні і комфортні у використанні. Більшість товарів та послуг класифіковані за категорією, обмежена кількість оголошень в кожній рубриці, це дає можливість звільнити застосунок від спаму та інформаційного сміття [2].

1.1.1 Опис процесу діяльності

У системі можна виділити три ролі : адміністратор, зареєстрований користувач (клієнт) і незареєстрований користувач (гість). Користувач може переглядати і купувати товари, але розміщувати оголошення і продати товар має право тільки зареєстрований користувач. Адміністратор верифікує оголошення та може допустити або не допустити його публікацію. Після заповнення клієнтом всіх необхідних полів при додаванні оголошення, воно надходить на верифікацію, тобто перевірку його коректності згідно з правилами користування системою. Якщо в оголошенні не виявлено жодних порушень воно публікується на дошці оголошень. В іншому разі оголошення не активується і клієнту приходить сповіщення про відмову.

Усі користувачі можуть переглядати активні оголошення, а також обирати за заданими критеріями та рейтингом. А згодом, зв'язатись з автором оголошення по контактному телефону, поданому в оголошенні або через чат додатку. Після продажу, або виходу терміну дії, оголошення деактивується.

1.1.1 Опис функціональної моделі

Опишемо функціональну модель системи наступним чином.

Акторами системи є:

- адміністратор;
- клієнт;
- незареєстрований користувач.

Адміністратор. Співробітник, який має змогу вносити зміни до бази даних, верифікувати оголошення та зареєстрованих користувачів.

Клієнт. Зареєстрований користувач на сайті, має змогу додавати, коментувати оголошення.

Незареєстрований користувач. Незареєстрований користувач (гість), може переглядати оголошення на сайті а також бачити контакти клієнтів.

Виділимо наступні основні варіанти використання:

- перегляд оголошень:
 - 1) переглянути всі доступні оголошення, розміщені на сайті;
 - 2) перегляд новинок: система видає новинки та популярні оголошення першочергово;
 - 3) пошук оголошень за заданими параметрами.
- розміщення оголошень :
 - 1) реєстрація – заповнення реєстраційної форми для збереження даних в системі
 - 2) додавання оголошення – заповнення форми, надання опису, фотографій та контактних даних.

Представимо структурну модель системи у вигляді схеми структурної варіантів використання, що наведена у графічному матеріалі.

1.2 Огляд наявних аналогів

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

До системи, що розробляється було проведено огляд наявних аналогів. Серед яких було виявлено наступні: OLX, Work.ua, AliExpress, E-bay та Amazon.

OLX. Розробники сайту створили платформу для купівлі або продажу товарів і послуг онлайн як приватним особам, так і представникам бізнесу. Користувачі OLX.ua можуть розміщувати свої оголошення, попередньо зареєструвавшись за допомогою мобільного телефону, електронної пошти або увійти через акаунти соціальних мереж. У формі подання оголошення необхідно додати опис товару, що пропонується, фотографії та свої контактні дані. Також у цьому додатку є можливість відгукнутися на вже опубліковане оголошення, зв'язавшись з його автором через онлайн-чат на сайті або за номером телефону. Оголошення на цьому сайті відображаються за датою їх розміщення. Вперше мобільні додатки для пристроїв на ОС Android і iOS сайт запустив влітку 2013 року. 2018 року додаток налічував 10 млн встановлень і відповідно велику кількість користувачів. У травні 2017 року сайт запустив сервіс доставки. Ця послуга передбачає блокування грошей, що мають бути сплачені, на карті покупця і їх списання тільки після отримання товару. Проект реалізовано з «УАРАУ» і «Нова пошта». Аналоги такого сервісу діють на багатьох дошках оголошень, таких як AliExpress, Amazon.com і eBay. Подібним до моєї розробки є дошка оголошень OLX. Але оскільки AdvertTech розрахований на іноземних користувачів, на відміну від українського аналогу, а також на підтримку лізинг-програм і можливість великої кількості вакансій роботи за кордоном, продажу нерозмитнених авто, він дає можливість розширити аудиторію користувачів до Європейського масштабу. Також очевидним плюсом моєї розробки є те, що вона цілком безкоштовна, на відміну від інших дошок оголошень та не лімітує кількість оголошень від одного користувача[3].

eBay — інтернет-компанія, що створена америнськими розробниками, вона керує однойменним сайтом eBay.com, також компанія створила онлайн

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

майданчик для проведення аукціонів і торговельний веб-сайт, на якому користувачі можуть публікувати свої товари та послуги. eBay.com має міжнародні локалізації, тобто співпрацює з іншими країнами[4].

eBay Inc. також належить міжнародна система, що здійснює електронні платежі через PayPal, сервіси VoIP та інші сервіси. У березні 2010 року компанія придбала 49 % акцій платформи електронної комерції Magento. Ebay також є певним аналогом мого додатку, але відрізняється тим, що у ньому немає можливості розміщувати оголошення послуг та роботи[5].

Amazon.com — один із перших інтернет-засобів, що орієнтовані на реалізацію реальних товарів загального попиту, це одна з найбільших світових компаній, що реалізує товари та послуги через Інтернет мережу. Компанія веде експансію на ринки інтернет-торгівлі в усьому світі. Станом на 2017 рік Amazon має окремі сайти для роздрібною торгівлі у Сполучених Штатах Америки, Франції, Канаді, Німеччині, Італії, Великій Британії та північній Ірландії, Іспанії, Нідерландах, Австралії, Бразилії, Японії, Китаї, Індії та Мексиці[10].

Work.ua — сайт пошуку роботи в Україні, його щомісяця відвідують 3,8 млн людей, більше 155 тис. — розміщують вакансії. За даними на серпень 2017 року на сайті зібрано близько 2 млн резюме та 90 тисяч вакансій. Основні можливості: пошук вакансій, створення резюме, надсилання резюме на вакансії, розсилка вакансій електронною поштою. Додаткові інформаційні розділи: статті про роботу та працевлаштування, новини в сфері HR, статистика зарплат, календар професійних свят, добірка гумору про роботу.

У сайту є можливість для сторонніх сайтів: налаштовувати на своїх сторінках трансляцію вакансій та статистику зарплат[7].

AliExpress.com — Інтернет-магазин роздрібною торгівлі малого бізнесу Китаю, пропонує товари міжнародним покупцям в Інтернеті. Належить групі компаній Alibaba. AliExpress — це найпопулярніший веб-сайт електронної торгівлі в Росії та був 10-м найпопулярнішим веб-сайтом у Бразилії. Сайт

допомагає китайським кустарним виробництвам поширювати свою продукцію клієнтам по всьому світу і як на Amazon або ж eBay, на AliExpress можна знайти практично все що завгодно для продажу.

Аналогічно до eBay, продавцями на Aliexpress можуть бути як компанії, так і приватні особи. Amazon має відмінності від Aliexpress, оскільки він працює лише як платформа для всесвітньої інтернет торгівлі, а не продає свій товар напряму споживачам. Aliexpress напряму з'єднує китайських підприємців із споживачами. На відміну від китайського Інтернет-магазину Taobao, продажі Aliexpress спрямовані насамперед на міжнародних покупців переважно із США, Росії, Бразилії чи Іспанії. Alibaba використовує AliExpress для того щоб розширити свою зону споживачів за межі Азії та конкурувати з світовими лідерами онлайн торгівлі як Amazon чи eBay. Щоб віднайти нових клієнтів AliExpress використовує партнерську програму.

Порівняння з аналогами наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння аналогів

Продукт	Інтернаціональність	Можливість розміщувати вакансії	Контакти з продавцями	Розмістити оголошення
Advert	+	+	+	+
E-bay	+	-	+	-
Amazon	+	-	-	-
OLX	-	+	+	+
Work.ua	-	+	+	+
AliExpres	+	-	+	-

Формування призначення і цілей створення системи :**1.3.1 Призначення розробки**

Розроблення веб – додатку дошки оголошень орієнтованої на іноземних замовників(роботодавців, продавців, продавців послуг).

1.3.2 Цілі та задачі розробки

AdvertTech призначена для спрощення входу на європейський ринок, поширення лізингу, шляхом усунення бар'єрів доступу вітчизняних і іноземних користувачів до ринку товарів та послуг.

Для досягнення поставленої мети потрібно вирішити наступні задачі :

- ведення оголошень;
- ведення користувачів інформаційної системи;
- перегляд оголошень;
- додавання оголошень зареєстрованими користувачами;
- редагування та видалення власних оголошень;
- перегляд категорій;
- розмежування доступу до інформації.

Висновок до розділу

Отже, був проведений опис предметного середовища, визначені вимоги, було знайдено аналоги та зроблено порівняльну характеристику, а також було визначено призначення розробки, цілі та задачу для досягнення цілей. Виходячи з порівняльної характеристики, можемо зробити висновок, що наше застосування буде корисним та більш інформативним, ніж аналоги.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Первісні дані вводяться в систему користувачем через веб - сервіс.

Дані по користувачу:

- логін ;
- пароль;
- ім'я клієнта в ситемі;
- номер телефону;
- електронна пошта.

Дані по оголошенню:

- назва оглошення;
- категорія;
- користувач – користувач, який додає оголошення ;
- місце розташування – місце розташування можна додати

зпоставивши точку на карті ;

- вартість;
- опис товару;
- зображення.

2.2 Вихідні дані

За результатами оформлене оголошення з'являється на сайті з наступними вихідними даними :

- назва оглошення;
- категорія;
- користувач – користувач, який додає оголошення ;
- місце розташування – місце розташування можна додати

зпоставивши точку на карті ;

- вартість;
- опис товару;

- зображення;
- телефон продавця;
- електронна пошта продавця;
- категорія товару;
- стан;
- термін початку показу оголошення;
- термін кінця показу оголошення.

2.3 Опис структури бази даних

Таблиця 2.1 – Таблиця користувачів

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
USER – таблиця користувачів сайту	Id	integer	Код, під яким користувач зареєстрований в базі даних
	username	Varchar255	Логін користувача
	phonenumber	Varchar255	Контактний телефон
	email	Varchar255	Електронна пошта користувача
	Firstname	Varchar255	Ім'я користувача
	Lastname	Varchar255	Прізвище користувача
	role	Varchar255	Роль користувача
	Access_token	Varchar255	Служить для визначення прав користувача
	Auth_key	Varchar255	Служить для ідентифікації користувача
	password	Varchar255	Пароль

Таблиця 2.2 – Таблиця об’яв

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
assortiment – таблиця об’яв сайту	Id	integer	Код, під яким об’ява внесена в базу даних
	Name_ru	text	Назва оголошення українською
	Name_et	text	Назва оголошення російською
	Name_en	money	Назва оголошення англійською
	Category_id	text	Зовнішній ключ категорії
	User_id	int	Зовнішній ключ користувача
	Rating	Int	Рейтинг оголошення
	Shown	Set(yes/no)	Видимість
	Email	Varchar255	Пошта продавця
	Price	float	Ціна
	Price_old	float	Стара ціна
	phone	Varchar255	Контактний телефон
	Description_ru	Text	Опис українською
	Description_et	Text	Опис російською
	Description_en	text	Опис англійською

Таблиця 2.3 – Таблиця категорій

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
categories – таблиця товарів сайту	Id	integer	Код, під яким товар внесений в базу даних

Продовження таблиці 2.3

	name	text	Назва категорії
	parent_id	integer	Послання на таблицю categories

Таблиця 2.4 – Таблиця груп

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
collaborate– таблиця товарів сайту	Id	integer	Код, під яким група внесена в базу даних
	Word	text	Назва групи
	assortiment_ID	integer	Послання на таблицю assortment
	User_id	integer	Посилання на таблицю user

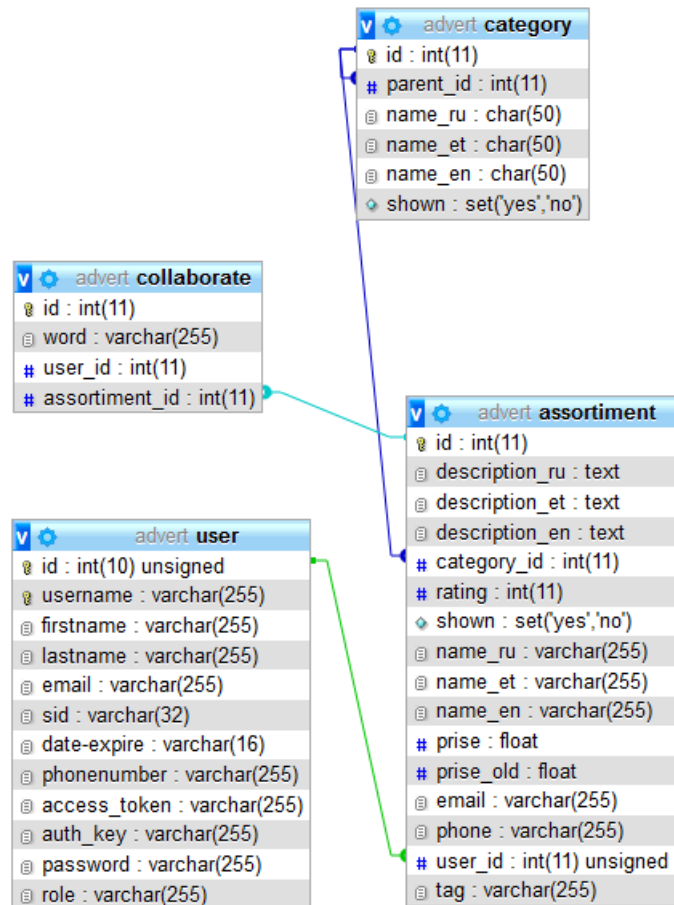


Рисунок 2.1 – Схема бази даних

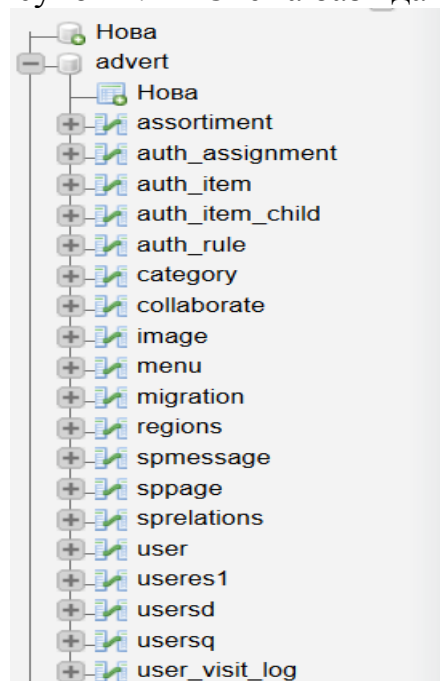


Рисунок 2.2 – Схема структурна бази даних

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id	int(10)	UNSIGNED	Hi	Немає		AUTO_INCREMENT
<input type="checkbox"/>	2	username	varchar(255) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	3	firstname	varchar(255) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	4	lastname	varchar(255) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	5	email	varchar(255) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	6	sid	varchar(32) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	7	date-expire	varchar(16) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	8	phonenumber	varchar(255) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	9	access_token	varchar(255) utf8_general_ci		Так	NULL		
<input type="checkbox"/>	10	auth_key	varchar(255) utf8_general_ci		Так	NULL		
<input type="checkbox"/>	11	password	varchar(255) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	12	role	varchar(255) utf8_general_ci		Hi	user		

Рисунок 2.3 - Таблиця user

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	1	id	int(11)		Hi	Немає	ID	AUTO_INCREMENT
<input type="checkbox"/>	2	description_ru	text utf8_unicode_ci		Hi	Немає		
<input type="checkbox"/>	3	description_et	text utf8_unicode_ci		Hi	Немає		
<input type="checkbox"/>	4	description_en	text utf8_unicode_ci		Hi	Немає		
<input type="checkbox"/>	5	category_id	int(11)		Hi	Немає		
<input type="checkbox"/>	6	rating	int(11)		Hi	Немає		
<input type="checkbox"/>	7	shown	set('yes', 'no') utf8_unicode_ci		Hi	yes		
<input type="checkbox"/>	8	name_ru	varchar(255) utf8_unicode_ci		Hi			
<input type="checkbox"/>	9	name_et	varchar(255) utf8_unicode_ci		Hi	nimi eesti keeles		
<input type="checkbox"/>	10	name_en	varchar(255) utf8_unicode_ci		Hi	name in english		
<input type="checkbox"/>	11	prise	float		Hi	Немає		
<input type="checkbox"/>	12	prise_old	float		Так	NULL		
<input type="checkbox"/>	13	email	varchar(255) utf8_general_ci		Hi	Немає		
<input type="checkbox"/>	14	phone	varchar(255) utf8_general_ci		Так	NULL		
<input type="checkbox"/>	15	user_id	int(11)	UNSIGNED	Hi	Немає		

Рисунок 2.4 – Таблиця assortment

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/> 1	<u>id</u>	int(11)			Ні	Немає	ID	AUTO_INCREMENT
<input type="checkbox"/> 2	<u>parent_id</u>	int(11)			Так	NULL	Родительская категория	
<input type="checkbox"/> 3	<u>name_ru</u>	char(50)	utf8_unicode_ci		Ні	Название категории	Название	
<input type="checkbox"/> 4	<u>name_et</u>	char(50)	utf8_unicode_ci		Ні			
<input type="checkbox"/> 5	<u>name_en</u>	char(50)	utf8_unicode_ci		Ні	Category name	Название по английски	
<input type="checkbox"/> 6	<u>shown</u>	set('yes', 'no')	utf8_unicode_ci		Ні	yes	Категория отображается	

Рисунок 2.5 – Таблица categories

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/> 1	<u>id</u>	int(11)			Ні	Немає		AUTO_INCREMENT
<input type="checkbox"/> 2	<u>word</u>	varchar(255)	utf8_general_ci		Ні	Немає		
<input type="checkbox"/> 3	<u>user_id</u>	int(11)			Ні	Немає		
<input type="checkbox"/> 4	<u>assortiment_id</u>	int(11)			Ні	Немає		

Рисунок 2.6 – Таблица collaborate

Висновок до розділу

Отже, було проаналізовано та представлено ER-модель, таблиці бази даних та лише структуру, необхідну для функціонування основних функцій системи, а також проілюстровано які дані, та у якому форматі вони зберігаються в базі. Вказано які є первинні ключі, автоінкременти а також вторинні ключі та залежності між сутностями. База даних розроблена за допомогою MySQL і створювалась безпосередньо на сервері за допомогою PhpMyAdmin. Усі запити та процедури реалізовані за допомогою php коду у пошукових моделях. Розгорнута модель бази даних наведено у графічному матеріалі.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Колаборативна фільтрація, або спільна фільтрація – це підхід, що знайшов своє місце у чималій кількості сучасних рекоєндаційних систем. Якщо сказати просто, то колаборативна фільтрація - вибірка інформації, в моєму випадку оголошень за подібністю запитів користувачів, а також на основі побіності товарів між собою. Колаборативна фіільтрація приміняється для роботи з великими масивами даних. Ці методи були корисними, при дослідженні та аналізі великих масивів даних, які виникають велектронній торгівлітавеб-додатках, що зосереджуються на даних користувача, іт.д.У більш новому, вузькому сенсі колаборативна фільтрація - це один з методів побудови передбачення в системах рекомендацій, в основу якого покладена оцінка та рейтинг окремого оголошення. Основний постулат колаборативної фільтрації: хто однаково оцінювали будь-які оголошення в минулому, схильні давати схожі оцінки подібних оголошень і в майбутньому. Наприклад, при кодаборативній фільтрації список його переваг. Прогнози складаються для кожного користувача, хоч і статистика збирається з усіх зареєстрованих користувачів.

Зокрема увага зосереджена на колаборативній фільтрації даних, призначених для відвідувача, що хоче отримати рекомендацію, само і у багатьох інших випадках. У більш вузькому розумінні, колаборативна фільтрація—це один з методів побудови передбачення в рекомендаційних системах, який за основу бере оцінку та рейтинг окремого оголошення

Якщо говорити про сучасне, більш вужче значення то це один з методів побудови прогнозу в системах, що призначені для рекомендації, який використовує відомі переваги, їх ще називають оцінками, конкретної групи користувачів для прогнозування невідомих переваг користувачів іншої групи.

Вона має місце, ажде нові предмети або користувачі являють собою велику проблему для рекомендаційних систем. Частково цю проблему допомагає вирішити підхід, в основі якого лежить аналіз вмісту, так як він опирається не на оцінки, а на атрибути, що в свою чергу допомагає включати нові предмети в рекомендації для користувачів. Проте проблему з наданням цих рекомендацій для нового користувача вирішити стає складніше.

Якщо говорити про сучасне, більш вузке значення то це один з методів побудови прогнозу в системах, що призначені для рекомендації, який використовує відомі переваги, їх ще називають оцінками, конкретної групи користувачів для прогнозування невідомих переваг користувачів іншої групи. Головним припущенням цього методу: ті, хто однаково оцінювали будь-які об'єкти в минулому, схильні давати в більшій мірі схожі оцінки інших предметів і в майбутньому. Наприклад, при використанні даного алгоритму в якомусь музичному застосунку, музика сподобається користувачеві, маючи неповний список його симпатій та антипатій. Рекомендації формуються окремо для кожного користувача, враховуючи його попередній вибір. Це є відмінною рисою колаборативної фільтрації від більш простих підходів, дає середню оцінку для кожного об'єкта інтересу, наприклад того, будується на кількості людей, що проголосували, чи оцінили даний предмет і, звісно, це є її значною перевагою

Основні проблеми колаборативної фільтрації . Розрідженість даних. Рекомендаційні системи, зазвичай, будуються на тому, що вони містять у собі великі за обсягом масиви користувачів та оголошень, або товарів, чи послуг. Але часто так трапляється, що особи, яким сподобався товар, чи вони придбали його не завжди ставлять оцінку. І як наслідок маємо досить розріджені дані в матриці "користувач-оголошення". Це є надзвичайно великою проблемою для молодих рекомендаційних систем. І саме це призводить до проблеми, так званого «холодного старту».

Різноманітність. У колаборативній фільтрації спершу необхідно збільшити асортимент та багатогранність. Але є деякі алгоритми, що можуть ускладнити ситуацію зокрема в продажах та послугах, як це часто трапляється в інтернет магазинах дуже складні умови для просування продукції нових, маловідомих брендів, так як їх витісняють широковідомі бренди, що давно перебувають на ринку. А це призводить до того, що відомі розрекламовані бренди не дають дорогу новим альтернативним маловідомим брендам, що подають надії. і приводить до одноманітності ринку. Білі ворони.

«Білі ворони» - це такі користувачі, вибір яких практично ніколи не можна спрогнозувати, тому що він у більшості випадків не співпадає з думкою більшості. Але, оскільки, таких людей небагато, то науковці припинили вдосконалення алгоритму, у цьому напрямку.

Синонімією називається тенденція схожих або однакових предметів мати різні імена. Це є негативним чинником, бо амйже всі пошукові системи та система рекомендацій просто не «бачать» аналогію між іменами предметів і тому ставляться до них як до різних. Наприклад, телефон і смартфон система не ототожнює, а бачить як різні об'єкти

3.2 Алгоритми колаборативної фільтрації.

Алгоритм колаборативної фільтрації Item-Base - це такий різновид колаборативної фільтрації, що система рекомендує користувачеві товари, які подібні до тих, що він раніше переглядав.

Характеристикою користувача A є переглянути ним оголошення. Для цього оголошення з переглянутих/вибраних, береться m сусідніх оголошень, тобто відбирається m найбільш подібних відносно переглядів/ оцінок користувачів. При побудові системи рекомендацій для мультфільмів, m приймає значення від 0 до 5. Всі об'єкти-сусіди об'єднують в сукупність, з якої відкидаються елементи, які переглянув, або оцінив пеній користувач, наприклад це буде користувач A . А з залишку подібних елементів

формується топ n рекомендацій. Отже, при item-based методі у укладанні рекомендацій участвують всі користувачі, які вподобали той чи інший елемент. А також був задіяний метод User-based підхід. Він полягає в тому, що користувачу можуть подобатись товари, які обрали інші, схожі на нього, користувачі.

Обидва алгоритми ідентичні за обрахунком

$$\hat{r}_{ui} = \bar{r}_i + \frac{1}{\sum_{i' \in R(u)} |\text{sim}(i, i')|} \sum_{i' \in R(u)} \text{sim}(i, i') (r_{u,i'} - \bar{r}_{i'}) \quad (3.1)$$

Міра подібності ж $\text{sim}(u, u')$ обраховується за матрицею оцінок R :

$$\text{sim}(u, u') = \frac{\sum_{i \in R(u) \cap R(u')} (r_{u,i} - \bar{r}_u) (r_{u',i} - \bar{r}_{u'})}{\sqrt{\sum_{i \in R(u) \cap R(u')} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in R(u) \cap R(u')} (r_{u',i} - \bar{r}_{u'})^2}} \quad (3.2)$$

У наборах оголошень поданих у вигляді векторів відзначені пошукові критерії, товари, що мають найбільшу подібну категорію, ціну, а також кількість переглядів, тобто самі популярні. Саме так формується рейтинг для кожного товару. Алгоритм на основі діяльності користувача із схожими пошуковими запитам та переглядами формує цільову пропозицію, які, на думку системи будуть задовольняти користувача. Основна ціль даного сервісу зі сторони додатку: збільшити пропозицію користувачеві, а також на основі підходу Item-based рекомендації формуються і пропонуються системою на основі підходу User-based. Інформація, що буде використана для формування рекомендацій:

- вік користувача ;
- регіон, в якому подане оголошення;
- категорія оголошень;
- ціна, указана в оголошенні.

Для отримання даних, що беремо оцінки користувача. Значення оцінки можуть бути від 0 до 5. виставляється автоматично рейтинг «1», у випадку, коли оголошення переглядається користувачем за його запитом, то оцінка становить «5». Побудуємо вектор з усіх наявних критеріїв, що описує

оголошення, та застосуємо його для кластеризації. Також у вектор буде входити топ 20 елементів певного виду заснований на кількості оцінок користувачів цієї системи, наприклад топ 10 пральних машин LG, чи топ 10 пральних машин по місту Київ, що отримали найвищі оцінки користувачів. Даний вибір заснований на тому, що користувача на основі ставлення до найбільш популярних пральних машин можна віднести до певної групи.

3.2 Математична модель

Основною задачею є надання користувачу найбільш підходящих цільових пропозицій. Постановка задачі матиме наступний вигляд:

Дано:

- множина оголошень – U ;
- кількість оголошень – n ;
- вектор оголошення з яким виконується порівняння – u' , де $i = \overline{1, n}$;
- кількість товарів – m ;
- конкретний товар – j , де $j = \overline{1, m}$.

Змінні:

$r_{u,j}$ - передбачувана оцінка оголошення;

$r'_{u,j}$ - реальна оцінка оголошення;

$sim(u, u')$ - функція подібності оголошень.

Цільова функція:

$$z = \frac{1}{m} \sum_j^m 1 - \frac{|r_{u,j} - r'_{u,j}|}{\max(r'_{u,j}, r_{u,j})} \rightarrow \max \quad (3.3)$$

де

$$r_{u,j} = \sum sim(u, u') * r'_{u,j} \quad (3.4)$$

Обмеження:

$$0 < r_{u,j} < 5 \quad (3.5)$$

$$0 < r'_{u,j} < 5 \quad (3.6)$$

$$0 < sim(u, u') < 1 \quad (3.7)$$

Для вирішення даної задачі було обрано колаборативну фільтрацію і для визначення подібності оголошень обрано методи машинного навчання з подальшими записами в базу даних, тому що поєднавши з сучасними інструментами розробки на виході отримуємо збільшення швидкості виконання, і можливість поєднання оголошень у групи за подібністю категорій.

Тому опишемо ще математичну постановку задачі методу кластеризації.

Дана множина оголошень U , і кожна множина виступає вектором атрибутів. Необхідно відшукати функцію F , яка зможе розбити шукану множину на кластери C , і буде визначати імовірність належності оголошення певному кластеру $\mu(i)$

$$U = \{u_1, u_2, \dots, u_m\} \quad (3.6)$$

де n - кількість оголошень.

$$C = \{c_1, c_2, \dots, c_k, \dots, c_h\} \quad (3.7)$$

де h - кількість кластерів, яка в даному випадку не є відомою спочатку.

c_k – кластер, що об'єднує подібні елементи множини U .

$$c_k = \{u_i, u_g \mid u_j \in U, u_g \in U \text{ \& } d(u_i, u_g) < \sigma \}, \quad (3.8)$$

де – величина, яка визначає значення близькості для включення об'єкту до кластеру, $d(u_i, u_g)$ – відстань між векторами оголошень.

$$\mu_k(u_i) = d(u_i, u_k), \quad (3.9)$$

$$0 < \mu_k(u_i) < 1, \quad (3.10)$$

$$\sum_{k=0}^h \mu_k(u_i) = 1, \quad (3.11)$$

Ці обмеження показують, що предмет може відноситись до різних кластерів з різними степенями належності, і сума належності предмета до усіх кластерів рівна одиниці.

3.3 Огляд методів розв'язання

3.3.1 Огляд методів підготовки цільових пропозицій

Розглянемо два основні методи та їх різновиди рекомендаційних систем.

Порівняння методів надано в таблиці 3.1

Таблиця 3.1 – Порівняння методів підготовки цільових пропозицій

	Вхідні дані	Як використовується	Проблеми
Фільтрація на основі змісту	Об'єкти які користувач оцінив	Оцінюється схожість за параметрами об'єктів	Велика залежність від предметної області, ігнорування досвіду інших користувачів

Продовження таблиці 3.1

Колаборативна фільтрація на основі пам'яті	Оцінки виставлені користувачем; Оцінки об'єктів , виставлені іншими Користувачами	Знаходиться схожість між користувачами на основі їх оцінок	Проблема «холодного старту», проблема «білих ворон»
Колаборативна фільтрація на основі моделі (кластеризації)	Діяльність користувачів в Соціальних мережах. Оцінки об'єктів , виставлені іншими користувачами	Будуються кластери користувачів на основі їх векторів і поточному надаються рекомендації відповідно до оцінок всередині кластеру	Не враховуються вподобання
Гібридна Колаборативна Фільтрація	Оцінки виставлені користувачем; Оцінки об'єктів , виставлені іншими користувачами; Діяльність користувачів в соц. мережах.	Будуються кластери користувачів на основі їх векторів з врахуванням вподобань і поточному надаються рекомендації відповідно до оцінок всередині кластеру	Згадані вище проблеми відсутні

3.4 Алгоритм застосування модифікованого підходу

Опишемо алгоритм застосування модифікованого підходу у вигляді 11 кроків:

Крок 1. ОБЧИСЛИТИ вагові коефіцієнти для кожного оголошення, за схожістю критеріїв дослідження використовуючи міру схожості

Крок 2. ВИБРАТИ зі списку оголошення, які є найбільш наближеними до шуканого оголошення, групу визначеного розміру.

Крок 3. ОБЧИСЛИТИ прогнозований рейтинг елемента на основі оцінок оголошень з обраної групи .

Крок 4. ОБЧИСЛИТИ коефіцієнти ваги для кожного оголошення, на основі подібності властивостей використовуючи міру схожості

Крок 5. ВИБРАТИ зі списку оголошень, групу оголошень, що найбільшою мірою схожі до заданого, на основі вагових коефіцієнтів.

Крок 6. ОБЧИСЛИТИ прогнозовану оцінку оголошення на основі переглядів схожих

Крок 7. ОБЧИСЛИТИ вагові коефіцієнти для кожного користувача за схожістю їх пошукових запитів, використовуючи міру схожості

Крок 8. ВИБРАТИ зі списку користувачів, що найбільшою мірою подібні до даного користувача.

Крок 9. ОБЧИСЛИТИ очікуваний рейтинг оголошення на основі оцінок користувачів з обраної групи.

Крок 10. ОБЧИСЛИТИ вагові коефіцієнти для кожного з методів фільтрації .

Крок 11. ОБЧИСЛИТИ результуючий прогнозований рейтинг елементу.

На цьому робота алгоритму завершується.

3.5 Результати досліджень ефективності методу

Для проведення експерименту використовувались дані відкритим до них доступом. Та, щоб провести експерименти буде достатньо використати лише три значення :

- ідентифікатор оголошення;
- ідентифікатор користувача;
- оцінка оголошення користувачем.

Даний набір даних містить в собі інформацію про 100 товарів та оцінки 50000 користувачів в наступному вигляді (таблиця 3.2):

Таблиця 3.2 – Структура експериментальних даних

Ідентифікатор оголошення	Ідентифікатор користувача	Оцінка
1	1	5
145	3	2
3648	4	3
378	1	1
229	3	4
1	4	5

Оскільки оцінки – це по суті числа з інтервалу від 1 до 5, то в інформаційній системі можна їх представити у вигляді булевого маркеру – або користувачу сподобався товар, або ні, тобто позитивна, або ж негативна оцінка, тому має місце перетворення, наведене нижче :

$$r'/n=0,$$

$$\text{якщо } r' \in 1,2, r' \in N \text{ 1 ,}$$

$$\text{якщо } r' \in 3,5, r' \in N \text{ (2.12)}$$

Отже, якщо користувач u не задоволений елементом s , то його оцінка $r' \in 1,2$

і навпаки – сподобався, якщо його оцінка

$r' \in 3,5$.

Тобто, отримаємо дані наступного вигляду (таблиця 3.3):

Таблиця 3.3 – Структура експериментальних даних після перетворення

Ідентифікатор товару	Ідентифікатор користувача	Оцінка
1	1	1
145	3	0
3648	4	1
378	1	0
229	3	1
1	4	1

Дані необхідно розділити у співвідношення 75/ 25 відсотків :

З них тестових – 25%, а ті, що взяті для підрахунків – 75%

Тобто потрібно визначити оцінки, що дали об'єктам користувачі на основі даних для підрахунків, а згодом їх порівняти з тестовими. Завдяки цьому, можемо визначити ефективність алгоритму і релевантність передбачених оцінок.

В якості оцінок ефективності роботи алгоритмів будемо зазначимо наступні міри :

- точність;
- повнота;
- F – міра .

Ці міри можна визначити за формулами :

$$P = \frac{t_p}{f_p + t_p} \quad (3.13)$$

$$R = \frac{t_p}{f_p + t_p} \quad (3.14)$$

$$F = 2 * \frac{P * R}{P + R} \quad (3.15),$$

де P – точність;

R – повнота;

F – F-міра;

t – істинно-позитивні рекомендації (true-positive);

f – хибно-позитивні рекомендації (false-positive);

fK – хибно-негативні рекомендації (false-negative).

В цьому випадку:

- істинно-позитивні рекомендації (true-positive)
- оцінки що, є задовільними як і у результаті роботи алгоритму,

так і у тестових даних;

- хибно-позитивні рекомендації (false-positive)
- оцінки що, є незадовільними у тестових вибірках даних але є

задовільними у результатах роботи алгоритму;

- хибно-негативні рекомендації (false-negative)
- оцінки що, є задовільними у тестових вибірках даних але є

незадовільними у результатах роботи алгоритму.

Дослідження точності проведені з використанням різних комбінацій алгоритмів указаних вище :

- колаборативна фільтрація (CF);
- фільтрація на основі вмісту (CB);
- демографічна фільтрація (DF);
- CB + CF;
- CF + DF;
- DF + CB;
- модифікований підхід (CF+CB+DF).

Результати експериментів зображені у вигляді таблиці 3.4 та рисунку 3.1:

Таблиця 3.4 – Результати експериментів

Алгоритм	Точність	Повнота	F-міра
CF	0.621	0.58	0.588
CB	0.719	0.7	0.732
DF	0.534	0.64	0.62
CB+CF	0.631	0.642	0.62
CF+DF	0.601	0.62	0.607
DF+CB	0.67	0.687	0.694
Модифікований метод	0.736	0.73	0.75



Рисунок 3.1 – Залежність різних методів розв'язання

Висновок до розділу

Було порівняно різні методи підготовки цільових пропозицій і виявлено, що найефективнішими серед них є алгоритм демографічної фільтрації, а також модифікований метод колаборативної фільтрації. А також все це було реалізовано в алгоритмі колаборативної фільтрації, що використовується в програмному забезпеченні з використанням алгоритму, таблиці Collaborate в базі даних і за допомогою sql запитів.

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

4. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

При створенні програмного продукту були використані HTML, CSS, JavaScript, Php, My SQL, YiiFramework2 basic.

PHP- це мова програмування, що була створена спеціально для розробки веб-продуктів. PHP спочатку стояв для Personal Home Page, але тепер він означає рекурсивний ініціалізм PHP: Hypertext Preprocessor.

Код PHP може бути виконаний з інтерфейсом командного рядка (CLI), вбудованим в HTML-код, також використовується разом із веб-шаблонами Bootstrap і веб-фреймворками. PHP-код зазвичай обробляється інтерпретатором PHP, реалізованим як модуль у веб-сервері, або у вигляді виконуваного файлу CGI. Веб-сервер поєднує в собі роботу PHP-коду в поєднанні з html версткою, а також з створеною веб-сторінкою. PHP може використовуватися для багатьох завдань програмування поза веб-контекстом, таких як автономні графічні програми і робототехнічний контроль за безпілотними роботами .

Стандартний інтерпретатор PHP, який працює на Zend Engine, і є безкоштовним програмним додатком, створеним під керівництвом PHP. PHP був широко перенесений і може бути розгорнутий на більшості веб-серверів практично на будь-якій операційній системі і платформі безкоштовно .

Спершу, PHP без письмової специфікації, причому початкова реалізація виступала в якості де-факто стандарту, на який прагнули слідувати інші реалізації. З 2014 року продовжується робота над створенням офіційної специфікації PHP.

Інтерпретатор PHP виконує лише PHP-код у межах своїх роздільників. Все, що є поза тегами інтерпретатора, не обробляється , але текст, що не входить до складу PHP, все ще є об'єктом контрольних структур, описаних у

кодi PHP. Найбiльш поширенi роздiлювачi Закривати роздiли PHP. Укорочена форма

Перша форма роздiльникiв. В XHTML та iнших XML-документах, створює правильно сформованi iнструкцiї з обробки XML. Це означає, що отримана сумiш коду PHP та iншої розмiтки в файлi сервера сам по собi є якiсно сформованим XML.

Змiннi префiксуються символом долара, а тип не потрiбно вказувати заздалегiдь. PHP 5 запроваджує пiдказки типу, що дозволяє функцiям змушувати iх параметри бути об'єктами конкретного класу, масивiв, iнтерфейсiв або функцiй зворотного виклику. Однак перед PHP 7.0 пiдказки типу не можуть бути використанi з скалярними типами, такими як цiле число або рядок.

На вiдмiну вiд iменi функцiй та класiв, назви змiнних звертають увагу на регiстр. Рiвнi подвiйного цитування ("") i рядки heredoc забезпечують можливiсть iнтерполяцiї значення змiнної в рядок. PHP розглядає новi рядки як пробiли у виглядi мови вiльної форми, а оператори припиняються крапкою з комою. PHP має три типи синтаксису коментарiв: / * * / вiдмiтки блокiв i вбудованих коментарiв; // або # використовуються для однорядкових коментарiв. Оператор echo використовується в PHP для виведення тексту.

З точки зору ключових слiв i синтаксису мови, PHP схожий на синтаксис стилю C. якщо умови, для i в циклi, i функцiя повертається аналогiчно в синтаксисi до мов, таких як C, C ++, C #, Java i Perl.

Типи даних

PHP зберiгає цiлi числа в платформи-залежному дiапазонi, або 64-розрядне або 32-бiтове цiле число, еквiвалентне довгому типу C-мови. Невизначенi цiлi перетворюються в пiдписанi значення в певних ситуацiях; така поведiнка вiдрiзняється вiд iнших мов програмування. Цiлi змiннi можна призначати за допомогою десяткових (позитивних i негативних), вiсiмкових, шiстнадцяткових i двiйкових позначень.

Числа з плаваючою точкою також зберігаються в специфічному для платформи діапазоні. Вони можуть бути вказані з використанням нотації з плаваючою точкою або двох форм наукової нотації. PHP має рідний булевий тип, подібний до нативних булевих типів в Java і C ++. Використовуючи правила перетворення булевого типу, ненульові значення інтерпретуються як істинні і нульові як помилкові, як у Perl і C ++.

Null тип , що за замовчуванням позначає порожні дані; NULL - це єдине значення для заданого типу.

Змінні типу "ресурс" - посилання на ресурси з зовнішніх джерел. Вони, як правило, створюються функціями з конкретного розширення, і можуть оброблятися тільки функціями з одного розширення; приклади включають файли, зображення та ресурси баз даних.

Масиви можуть містити елементи будь-якого типу, які може обробляти PHP, включаючи ресурси, об'єкти і навіть інші масиви. Порядок зберігається в списках значень і в хешах з обома ключами і значеннями, і ці два можуть бути змішані. PHP розуміє рядки, що використовуються як і з подвійними, так і з одинарними кавичками, а також підтримує синтаксис 'heredoc' і 'nowdoc'.

Стандартна бібліотека PHP (SPL) намагається вирішити стандартні проблеми і впроваджувати ефективні інтерфейси і класи доступу до даних[9].

MySQL є системою управління реляційними базами даних з відкритим вихідним кодом (RDBMS). MySQL – вільне, з відкритим кодом забезпечення, згідно з GNU General Public License і є в доступу з іншими ліцензіями. MySQL була куплена компанією Sun Microsystems (нині Oracle Corporation). У 2010 році, коли компанія Oracle придбала Sun, компанія Widenius зробила популярним проект із відкритим кодом на виході MySQL, що спричинило створення MariaDB.

MySQL є складовою частиною програмного забезпечення LAMP (та інших), що є аббревіатурою для Linux, Apache, MySQL, Perl / PHP / Python.

MySQL задіяний у багатьох веб-додатках під керівництвом таких баз даних , як Drupal, Joomla, phpBB і WordPress . MySQL також використовується багатьма популярними веб-сайтами, включаючи Facebook, Twitter, Flickr, і YouTube.

MySQL написаний на C та C ++. Його SQL-парсер написаний у yacc, але він використовує домашній лексичний аналізатор. MySQL функціонує на великій кількості системних платформ, включно з AIX, BSDi, FreeBSD, HP-UX, eComStation, i5 / OS, IRIX, Linux, MacOS, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS / 2 Warp, QNX, Oracle Solaris, Symbian, SunOS, SCO OpenServer, SCO UnixWare, Sanos і Tru64. Також існує порт MySQL до OpenVMS.

Як серверне пз MySQL так і клієнтські бібліотеки використовують розподіл з двома ліцензіями. Вони пропонуються за версією GPL 2 або ліцензією на власність.

Підтримку можна отримати з офіційного посібника. Безкоштовна підтримка додатково доступна в різних каналах і форумах IRC. Oracle пропонує платну підтримку через продукти MySQL Enterprise. Вони відрізняються за обсягом послуг і в ціні. Крім того, існує ряд організацій третіх сторін, які надають підтримку та послуги, включаючи MariaDB та Percona.

MySQL має гарні відгуки, а рецензенти відмічають, що він "виконує дуже добре в середньому випадку", і що "інтерфейси розробників є, і документація (не кажучи вже про зворотний зв'язок у реальному світі через веб-сайти тощо) дуже, дуже добре ». Він також був протестований як "швидкий, стабільний і справжній багатокористувацький, багатопотоковий SQL-сервер баз даних".

MySQL — швидкий, зручний у використанні сервер баз даних, що має багатопотоковість. Основні характеристики якого : швидкість, зручність використання, великою стійкістю.

MySQL чудово підходить для невеликих програм та інформаційних систем де є підтримка багатопотоковості, що підвищує продуктивність системи в цілому, а також компілюється майже на усіх платформах, а найкраще сумісний з UNIX системами.

Для некомерційного використання MySQL є безкоштовним.

Переваги сервера MySQL:

- зручність і зрозумілість у інсталяції та використанні;
- надання необмеженої кількості користувачів одночасного доступу ;
- у таблицях, максимальна кількість рядків, що можна додати більше 50 мільйонів;
- оперативне виконання усіх команд;
- має дуже просту та ефективну систему безпеки[10].

Вперше розроблений Тімом Бернерсом-Лі у 1990 році, HTML є скороченням мови гіпертекстової розмітки. HTML використовується для створення електронних документів (так званих сторінок), які відображаються в World Wide Web. Кожна сторінка містить ряд з'єднань з іншими сторінками, які називаються гіперпосиланнями. Кожна веб-сторінка, яку ви бачите в Інтернеті, написана за допомогою однієї чи іншої версії HTML-коду.

HTML-код забезпечує правильне форматування тексту та зображень, щоб ваш веб-браузер міг відображати їх так, як вони мають намір виглядати. Без HTML, браузер не знає, як відображати текст як елементи або завантажувати зображення або інші елементи. HTML також надає базову структуру сторінки, на яку накладаються каскадні таблиці стилів, щоб змінити її зовнішній вигляд. HTML можна вважати кістками (структурою) веб-сторінки, а CSS - її шкірою (зовнішнім виглядом).

HTML — стандартна мова розмітки в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML

оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML складається з серії коротких кодів, введених автором сайту в текстовий файл - це теги. Потім текст зберігається як html-файл і переглядається через браузер, наприклад Internet Explorer або Netscape Navigator. Цей веб-переглядач читає файл і переводить текст у видиму форму, сподіваючись надати цю сторінку, як передбачав автор. Написання власного HTML-коду означає правильне використання тегів для створення свого бачення. Ви можете використовувати що-небудь від рудиментарного текстового редактора до потужного графічного редактора для створення HTML-сторінок. Теги - це те, що є звичайним текстом з HTML-коду. Можливо, ви знаєте їх як слова між кляцаннями. Вони дозволяють всі цікаві речі, такі як зображення, таблиці та інше, просто повідомляючи веб-переглядачу, що робити на сторінці. Різні теги виконуватимуть різні функції. Самі теги не відображаються, коли ви переглядаєте сторінку через веб-переглядач, але їхні ефекти. Найпростіші теги використовують лише застосування форматування до тексту, наприклад:

Ці слова будуть виділені напівжирним шрифтом , і їх не буде.

Їхній ефект буде полягати в тому, що текст, що міститься, буде виділений жирним шрифтом при перегляді через звичайний веб-браузер.

Документи HTML передбачають структуру вкладених елементів HTML. Вони вказуються в документі HTML-тегами.

У простому, загальному випадку, міра елемента позначена парою тегів: "стартовим тегом" і "кінцевим тегом". Текстовий вміст елемента, якщо такий є, розміщується між цими тегами.

Мітки також можуть містити додаткову розмітку тегів між початком і кінцем, включаючи суміш тегів і тексту. Це вказує на подальші (вкладені) елементи, як діти батьківського елемента.

Початковий тег може також містити атрибути всередині тегу. Вони вказують іншу інформацію, таку як ідентифікатори для розділів у документі, ідентифікатори, що використовуються для прив'язки інформації стилю до презентації документа, і для деяких тегів, таких як , що використовується для вбудовування зображень, посилання на ресурс зображення.

Деякі елементи, такі як перерва рядка, не дозволяють вбудований вміст, ні текст, ні інші теги. Вони вимагають лише одного порожнього тегу (схоже на стартовий тег) і не використовують кінцевий тег.

Багато тегів, зокрема кінцевий тег закриття для дуже часто використовуваного елемента абзацу, є необов'язковими. Браузер HTML або інший агент може зробити висновок про закриття кінця елемента з контексту і структурних правил, визначених стандартом HTML. Ці правила складні і не широко зрозумілі більшості кодерів HTML.

Таким чином, загальний вигляд елемента HTML: " content " . Деякі елементи HTML визначаються як порожні елементи і мають форму . Порожні елементи можуть не містити жодного вмісту, наприклад, тега

або вбудованого тега . Ім'я елемента HTML - це ім'я, яке використовується в тегах. Зауважте, що перед ім'ям кінцевого тегу має місце символ косої риски, а в порожніх елементах кінцевий тег не є ні обов'язковим, ні дозволеним. Якщо атрибути не згадуються, у кожному випадку використовуються значення за замовчуванням[11].

Каскадні таблиці стилів (CSS) - це такі таблиці, що необхідні для опису зовнішнього вигляду додатку , написаного на мові HTML, як правило використовується разом із технологіями JavaScript та html.

CSS проектувався для можливості коректного поділу змісту та відображення разом із макетом, різними шрифтами та кольорами. Завдяки цьому забезпечується доступ до контенту, гнучність системи, можливість керувати специфікацією відображення, а також запозичувати формати у інших сторінок, тільки вказавши потрібний CSS файл.

А розділене форматування та вміст дає можливість представити одну веб сторінку різними стилями і методами візуалізації, наприклад виводити текстовий вміст, чи представляти оголосом, адаптуватись під різні монітори, планшети та смартфони.

Каскадні імена беруться із схем пріоритетів, для визначення, яке правило використовується, чи скільки правил задіяно для відображення певного елемента відображення. І цей метод роботи каскадних пріоритетів є передбачуваним.

Специфікації CSS підтримуються консорціумом World Wide Web (W3C). Текст / css типу Інтернет-носія (тип MIME) зареєстрований для використання з CSS за допомогою RFC 2318 (березень 1998). W3C використовує безкоштовний сервіс CSS для перевірки документів CSS.

Крім HTML, інші мови розмітки підтримують використання CSS, включаючи XHTML, звичайний XML, SVG і XUL.

CSS має простий синтаксис і використовує ряд англійських ключових слів, щоб вказати назви різних властивостей стилю.

Таблиця стилів складається зі списку правил. Кожне правило або набір правил складається з одного або декількох селекторів і блоку оголошень.

Селектор

У CSS, селектори оголошують, яка частина розмітки стилю застосовується до відповідності тегів і атрибутів в самій розмітці.

Селектори можуть застосовуватися до наступного:

всі елементи певного типу, наприклад заголовки другого рівня h2

елементи, визначені атрибутом, зокрема:

id: ідентифікатор, унікальний у документі

class: ідентифікатор, який може коментувати декілька елементів у документі елементи в залежності від того, як вони розміщені відносно інших у дереві документів.

Класи та ідентифікатори чутливі до регістру, починаються з букв і можуть включати буквено-цифрові символи, дефіси та підкреслення. Клас може застосовуватися до будь-якої кількості екземплярів будь-яких елементів. Ідентифікатор може бути застосований лише до одного елемента.

Псевдо-класи використовуються в селекторах CSS для забезпечення форматування на основі інформації, яка не міститься в дереві документів. Одним з прикладів широко використовуваного псевдо-класу є: `hover`, який ідентифікує вміст тільки тоді, коли користувач «вказує» на видимий елемент, зазвичай, утримуючи курсор миші над ним. Він додається до селектора як `u: hover` або `#elementid: hover`. Псевдо-клас класифікує елементи документа, такі як: `link` або `: visited`, тоді як псевдоелемент робить вибір, який може складатися з часткових елементів, таких як `:: first-line` або `:: first-letter`[12].

Yii - це загальна структура веб-програмування, що означає, що вона може бути використана для розробки всіх видів веб-додатків, що використовують PHP. Завдяки своїй компонентній архітектурі та складній підтримці кешування, вона особливо підходить для розробки великомасштабних програм, таких як портали, форуми, системи управління контентом (CMS), проекти електронної комерції, веб-сервіси RESTful і так далі. Якщо ви вже знайомі з іншою системою, ви можете оцінити, як Yii порівнює:

Як і більшість фреймворків PHP, Yii реалізує архітектурний зразок MVC (Model-View-Controller) і сприяє організації коду на основі цього шаблону.

Yii приймає філософію, що код повинен бути написаний простим, але елегантним способом. Yii ніколи не намагатиметься надмірно розробити речі з метою суворо дотримуватися певного дизайну.

Yii - це повноцінний фреймворк, що забезпечує багато перевірених і готових до використання функцій: будівельників запитів і ActiveRecord як

для реляційних, так і для NoSQL баз даних; Підтримка RESTful API розвитку; підтримка багаторівневого кешування; і більше.

Yii надзвичайно розширюваний. Ви можете налаштувати або замінити майже кожен фрагмент коду ядра. Ви також можете скористатися повною архітектурою розширення Yii, щоб використовувати або розробляти розширення, які можна розповсюджувати.

Висока продуктивність завжди є головною метою Yii.

Yii - це не єдиний показник, він підтримується сильною командою розробників, а також великою спільнотою професіоналів, що постійно сприяють розвитку Yii. Команда розробників Yii уважно стежить за останніми тенденціями веб-розробки та найкращими практиками та особливостями, знайденими в інших рамках і проектах. Найбільш релевантні найкращі практики та функції, знайдені в іншому місці, регулярно включаються в основний фреймворк і піддаються через прості та елегантні інтерфейси[13].

4.2 Вимоги до технічного забезпечення

Архітектура технічних компонентів для роботи інформаційної системи отримується із можливості підтримувати необхідні операції процесу обслуговування, інтегрування в розроблені системи, безпеки роботи програми, виключення можливості хакерських атак і втрати даних. А також можливості розділеного доступу користувачем із відповідними правами.

Даний програмний продукт являє собою комплекс основних функцій процесу подачі оголошень та забезпечення контролю доступу до інформаційної системи, забезпечити конфіденційність даних, процедури розділеного доступу до системи.

Для коректної роботи інформаційної системи, до складу технічного забезпечення має входити :

- процесор з частотою такту не нижчу 1 гігагерц ;
- оперативна пам'ять об'ємом не менше ніж 256 мегабайт;

– інші складові технічного забезпечення можуть мати які завгодно параметри, якщо це значним чином не впливає на роботу системи;

Додатково має бути встановлене таке програмне забезпечення:

- операційна система Linux, Windows;
- браузер Mozilla, Chrome, Opera, Safari;
- Net Framework 3.5 і вище;
- Microsoft Word 2003/2007/2010.

Комп'ютерна периферія, до складу якої входить:

- монітор;
- мишка;
- клавіатура.

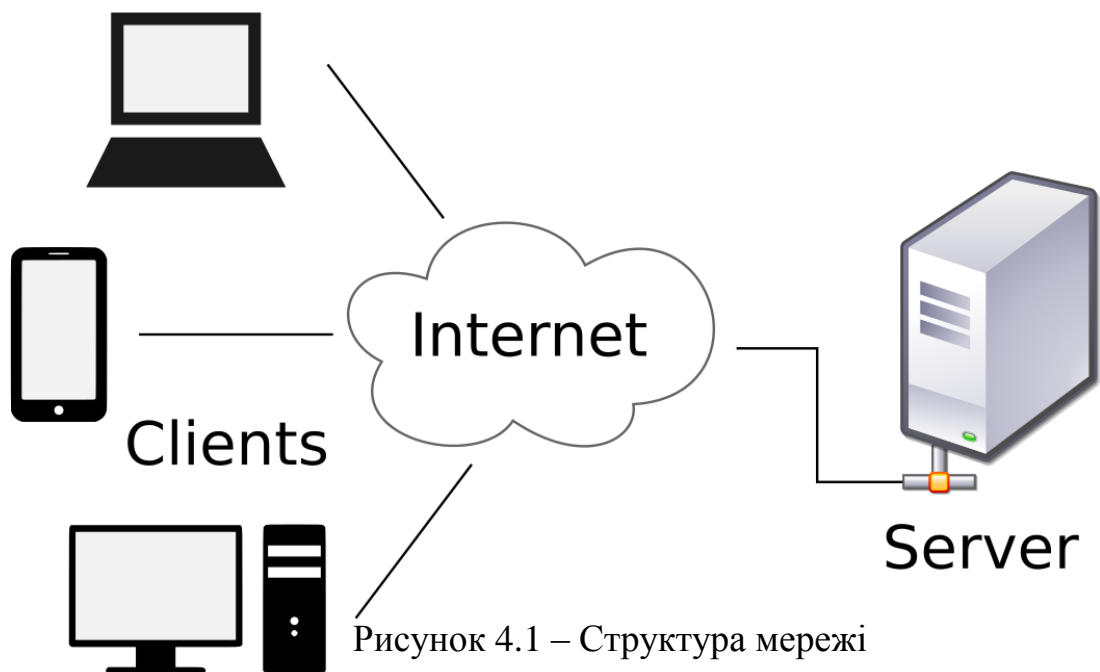


Рисунок 4.1 – Структура мережі

4.3 Діаграма компонентів

4.3.1 Особливості використання діаграми послідовності

Діаграма послідовностей просто зображує взаємодію між об'єктами в послідовному порядку, тобто порядок, у якому відбуваються ці взаємодії. Ми також можемо використовувати діаграми подій термінів або сценарії подій для посилання на діаграму послідовностей. Діаграми послідовностей

описують, як і в якому порядку об'єкти в системі функціонують. Ці діаграми широко використовуються бізнесменами та розробниками програмного забезпечення для документування та розуміння вимог до нових та існуючих систем.

4.3.2 Особливості використання діаграми класів

Для збереження множини і атрибутів, що постійно зберігаються, класу виокремлюється зберігання їх в таблиці бази даних. Тому звичайно можна спостерігати відповідність між сутностями таблиць бази даних та множиною класів. І відповідність асоціативних зв'язків між сутностями БД і класами.

4.3.3 Особливості використання діаграми компонентів

Якщо на діаграмі компонентів відобразити не лише двійкові виконувачі або бібліотечні файли, а всі необхідні для системи файли, включаючи файли допомоги, ініціалізації, масивів, тоді діаграма компонентів може називатись. Схема структурна послідовності наведена у графічному матеріалі.

4.4 Діаграма класів

Діаграма класів — статичне представлення структури моделі. Відображає статичні елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування.

На рисунку 5.1 представлена структурна схема класів, які відповідають за виконання таких функцій програми, як встановлення з'єднання з базою даних(каталогом товарів), робота з даними, додавання, видалення та редагування інформації в системі .

Діаграма містить сім класів, а саме:

- «Catalog Goods» - клас програми, відповідає за взаємодію програми з базою даних та містить основну інформацію про товари;
- «Composile Category» - клас, що відповідає за коректне з'єднання з базою даних та категоріями що містяться в базі, має функції додавання та редагування інформації що пов'язана з категоріями товарів;
- «Composile Goods» - клас, що відповідає за коректне з'єднання з базою даних та товарами що містяться в базі, має функції додавання та редагування інформації що пов'язана з інформацією про товари;
- «Category» - клас, функції якого виконують редагування категорій ;
- «Goods» - клас, що описує товари;
- «Advertisement» - клас призначенням якого є генерація оголошень.
- «User» - клас користувача, виконує аутентифікацію користувача і встановлює йому права. Схема структурна класів наведена у графічному матеріалі.

4.5 Діаграма компонентів

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі — під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

Приклад діаграми компонентів, які утворюють середовище для функціонування системи розміщення оголошень.

Підсистема WebStore містить три компоненти, пов'язані з онлайн-магазинами - Пошукова система, Кошик і Аутентифікація. Компонент Пошукова система дозволяє шукати або переглядати об'єкти, викриваючи наданий інтерфейс Пошук продукту, і використовує необхідний інтерфейс Пошук інвентаризації, наданий компонентом Inventory. Компонент кошика використовує інтерфейс керування замовленнями, наданий компонентом Orders під час перевірки. Компонент аутентифікації дозволяє клієнтам створювати обліковий запис, входити в систему, або виходити з системи і прив'язувати клієнта до деякого облікового запису. Підсистема обліку забезпечує два інтерфейси - керувати замовленнями та керувати клієнтами. Делегатні з'єднувачі пов'язують ці зовнішні контракти підсистеми з реалізацією контрактів за допомогою компонентів Замовлення та клієнтів. Підсистема складів забезпечує два інтерфейси пошуку та керування інвентаризацією, що використовуються іншими підсистемами та проводом через залежність. Схема структурна компонентів програмного забезпечення наведена у графічному матеріалі.

4.6 Специфікація функцій

Функції класів програмного забезпечення наведені в таблиці 4.9.

Таблиця 4.1 – Функції класів програмного забезпечення

Назва	Примітка
Клас: Layout – основний клас, який включає в себе функції управління інтерфейсом, роботою з базою даних, та ключає в себе виклики інших класів.	
public function behaviors()	Реалізує функціональну поведінку класу

Продовження таблиці 4.1

public static function tableName()	Виконує завантаження наявних даних із БД у програму
public function attributeLabels()	Ініціалізує дані на екран з таблиці БД
public function getTblAssortiments()	Виконує відображення даних з ьаблиці Assortiments
public function getParent()	Відображає батьківську категорію товару
public function getStructure()	Застосовується для відображення структурних зв'язків між батьківською і дочірніми категоріями
public function upload()	Застосовується для завантаження зображення до бази даних
public function getUser()	Застосовується для відображення інформації користувача
public function getCategory()	Застосовується для відображення інформації категорії
public function scenarios()	Застосовується для початку транзакцій у базі даних
public function search(\$params)	Функція пошуку за параметрами
public function getAssortiment()	Застосовується для відображення інформації товарів
public function contact(\$email)	Застосовується для зв'язу з адміністратором по пошті.
public function validatePassword(\$attribute, \$params)	Застосовується для валідації паролю користувачем.

Продовження таблиці 4.1

public function login()	Застосовується для відображення коректиних даних та верифікації даних для входу
public static function findIdentity(\$id)	Застосовується для ідентифікації користувача та відображення його порядкового номера з бази даних
public static function findIdentityByAccessToken(\$token, \$type = null)	Застосовується для ідентифікації користувача за типами доступу.
public static function findByUsername(\$name)	Застосовується для ідентифікації користувача та відображення його порядкового номера з бази даних
public function getId()	Застосовується для ідентифікації користувача та відображення його порядкового номера з бази даних
public function getAuthKey()	Застосовується для передачі хешованого паролю
public function validateAuthKey(\$authKey)	Застосовується для перевірки хешованого паролю
public function validatePassword(\$password)	Застосовується для валідації паролю користувачем.
public function similarityDistance(\$preferences, \$user1, \$user2)	Застосовується в колаборативному методі для визначення подібності особистісних даних користувачів
public function getRecommendations(\$preferences,	Застосовується для отримання і вирахування рекомендацій товару для користувача
public function actionIndex()	Застосовується для входу на головну сторінку застосування
public function actionView(\$id)	Застосовується для переходу на сторінку перегляду
public function actionCreate()	Застосовується для переходу на сторінку створення

Продовження таблиці 4.1

public function actionUpdate(\$id)	Застосовується для переходу на сторінку оновлення
public function actionDelete(\$id)	Застосовується для переходу на сторінку видалення
protected function findModel(\$id)	Застосовується для створення вибірки і ідентифікації моделі для оголошення
public function actions()	Застосовується для ініціалізації всіх функцій в контролері
public function actionLogin()	Застосовується для функціонування входу в систему користувачем
public function actionLogout()	Застосовується для функціонування виходу з системи користувачем
public function actionSignup()	Застосовується для функціонування модуля реєстрації
public function actionAddpost()	Застосовується для функціонування модуля подачі оголошення
public function actionUpload()	Застосовується для вивантаження і відображення зображення в оголошенні
public function actionContact()	Застосовується для функціонування модуля зв'язку з розробниками
public function actionAbout()	Застосовується для функціонування модуля «Про продукт»
public function actionEntry()	Застосовується для функціонування модуля подачі оголошення
public function actionComments()	Застосовується для функціонування модуля додавання коментарів
protected function isUserAuthor()	Застосовується для перевірки чи є користувач автором оголошення
public function actionStructure()	Застосовується для відображення структурних зв'язків

Висновок до розділу

Отже, в цьому розділі наведені програмні компоненти, та технічне забезпечення, зо було використане у процесі створення та написання AdvertTech, а також наведена специфікація функцій, що безпосередньо задіяні у роботі застосунку, але не наведені самі базові функції фреймворку, які задіяні в підключених класах Base, yii2-images та codemix.

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Для запуску програмного застосування необхідно в пошуковому рядку браузера прописати : “localhost:81/”

Головне вікно застосування містить головну сторінку, сторінку перегляду оголошень, категорій форму реєстрації та авторизації. 5.1.

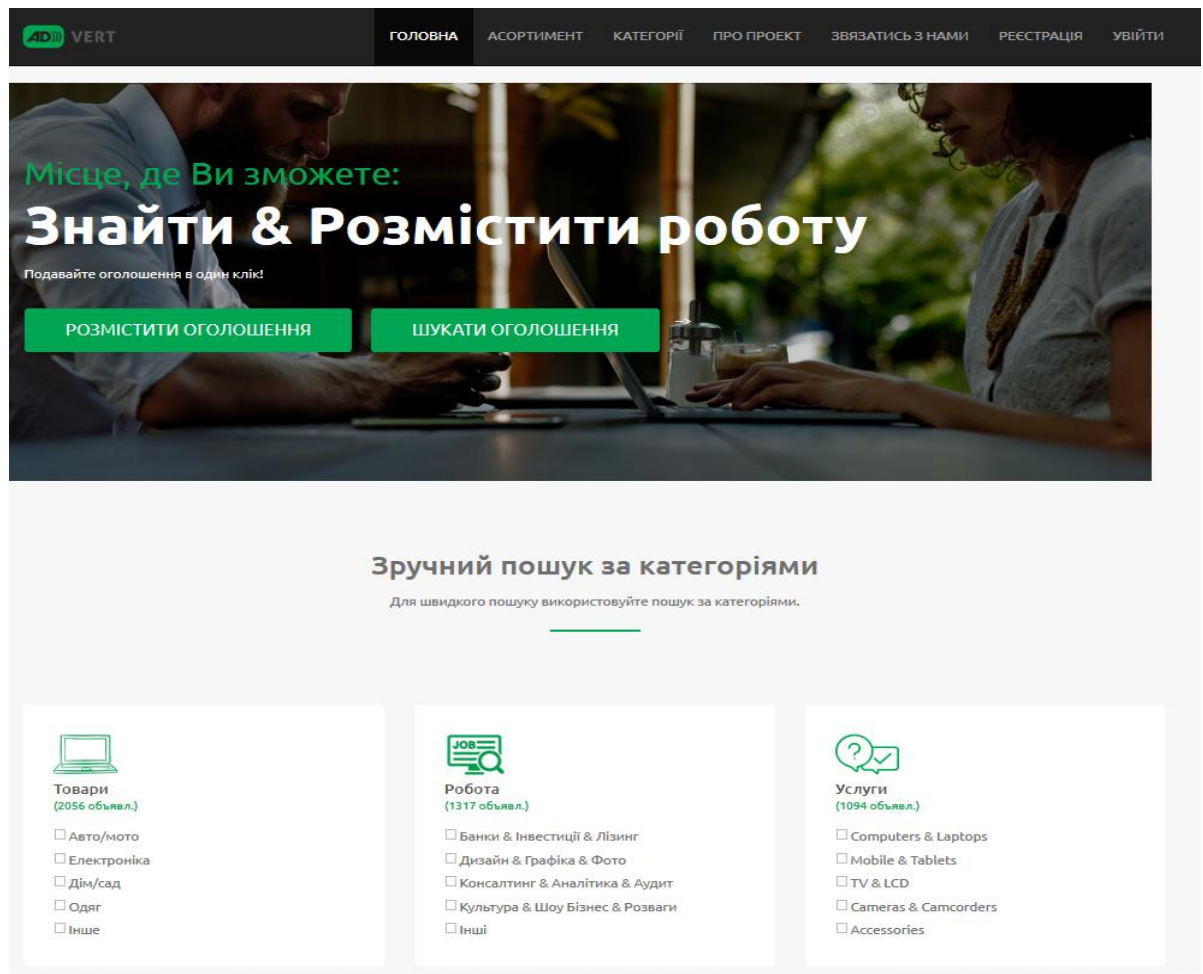


Рисунок 5.1 – Головне вікно інформаційної системи

Для авторизації потрібно обрати пункт меню увійти(рисунок 5.2). Там є можливість увести свої авторизаційні дані та увійти в особистий кабінет(рисунок 5.3).

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

AD VERT

ГОЛОВНА АСОРТИМЕНТ КАТЕГОРІЇ ПРО ПРОЕКТ ЗВ'ЯЗАТИСЬ З НАМИ РЕЄСТРАЦІЯ **УВІЙТИ**

Головна / Вхід

Вхід

Вхід

Логін

Необхідно заповнити «Логін».

Пароль

☒ Пам'ятати мене

Увійти

Рисунок 5.2 – Сторінка входу

AD VERT

ГОЛОВНА АСОРТИМЕНТ КАТЕГОРІЇ ПРО ПРОЕКТ ЗВ'ЯЗАТИСЬ З НАМИ РЕЄСТРАЦІЯ **УВІЙТИ**

Головна / Вхід

Вхід

Вхід

Логін

Пароль

☒ Пам'ятати мене

Увійти

Рисунок 5.3 – Введення параметрів підключення до БД

Далі, після натиснення на кнопку Увійти, переходимо на меню користувача 5.4.

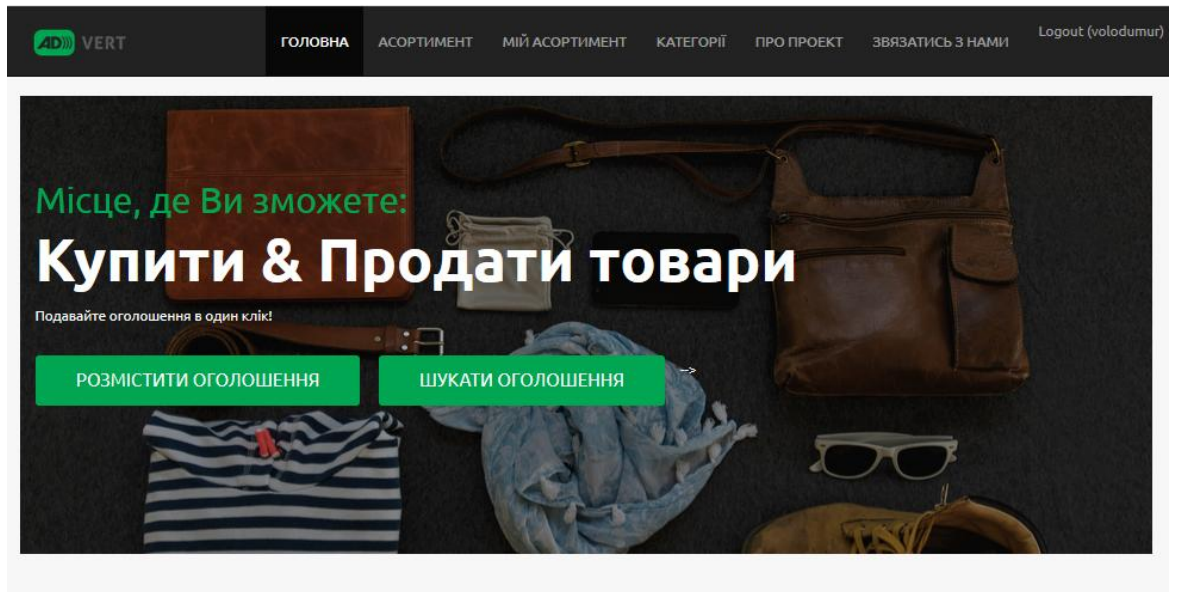


Рисунок 5.4 – Перехід на меню користувача

Натиснувши на кнопку Мій асортимент, з'явиться вікно (рисунок 5.5), де можна переглянути, редагувати або видалити власні оголошення

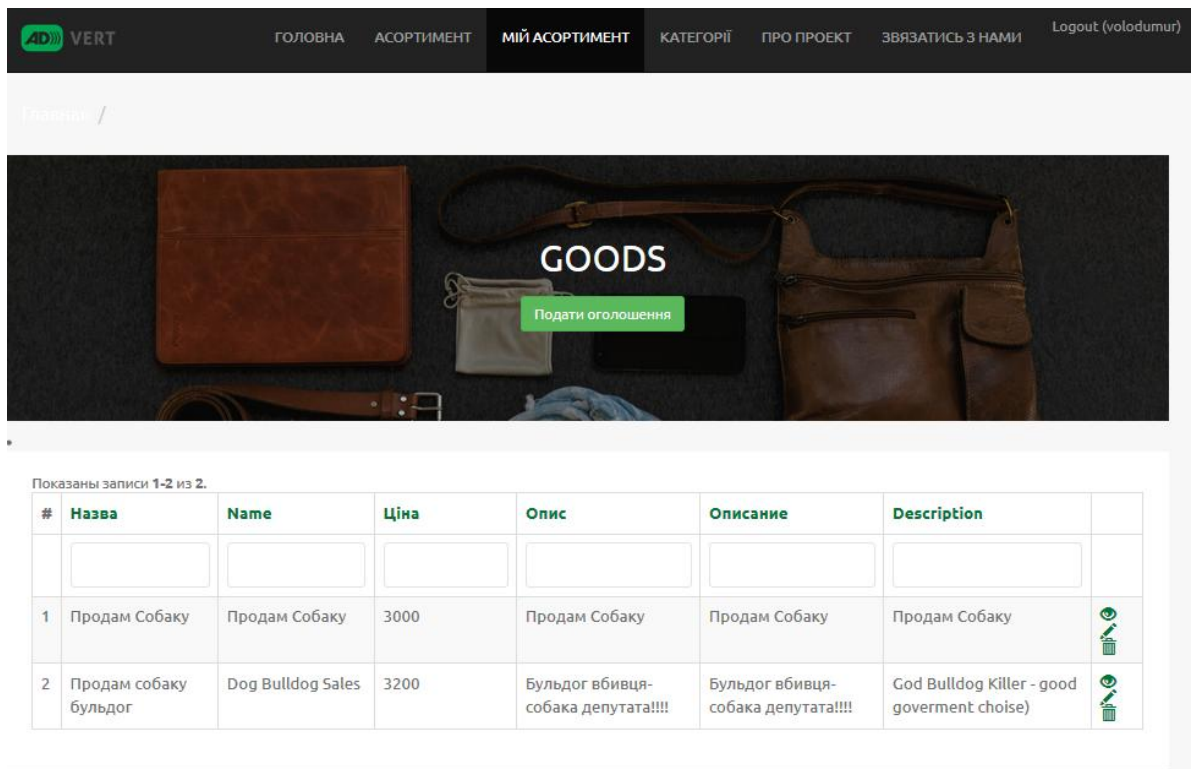


Рисунок 5.5 – Керування власними оголошеннями

Щоб додати оголошення необхідно натиснути кнопку Подати оголошення та з'явиться форма(рисунок 5.6). Необхідно заповнити форму даними(рисунок 5.7) та натиснути кнопку

название

Name

Рейтинг товару

Товар відображається

Ціна

Стара ціна

Пошта

Номер телефону

Опис

Описание

Description

Зображення

Огляд... Файл не вибрано.

Створити

Рисунок 5.6 – Форма подачі оголошення

Після введення усіх даних слід натиснути кнопку *Створити*. Тоді з'явиться оголошення на сайті (рисунок 5.7, рисунок 5.8). Якщо не всі поля були заповнені то підсвітяться червоним поля (рисунок 5.9). Якщо всі поля заповнені коректно, то оголошення з'явиться у загальному списку оголошень (рисунок 5.10).

Назва	Продаж морської свинки, пухнаста, весела, розмовляє, дуже активна
Название	Продаж морської свинки, пухнаста, весела, розмовляє, дуже активна
Name	Продаж морської свинки, пухнаста, весела, розмовляє, дуже активна
Рейтинг товару	5
Товар відображається	yes
Ціна	300
Стара ціна	
Пошта	vsevolodnestaiko@gmail.com
Номер телефону	+380967355233
Опис	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Описание	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Description	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою

Рисунок 5.7 – Інформація про замовлення

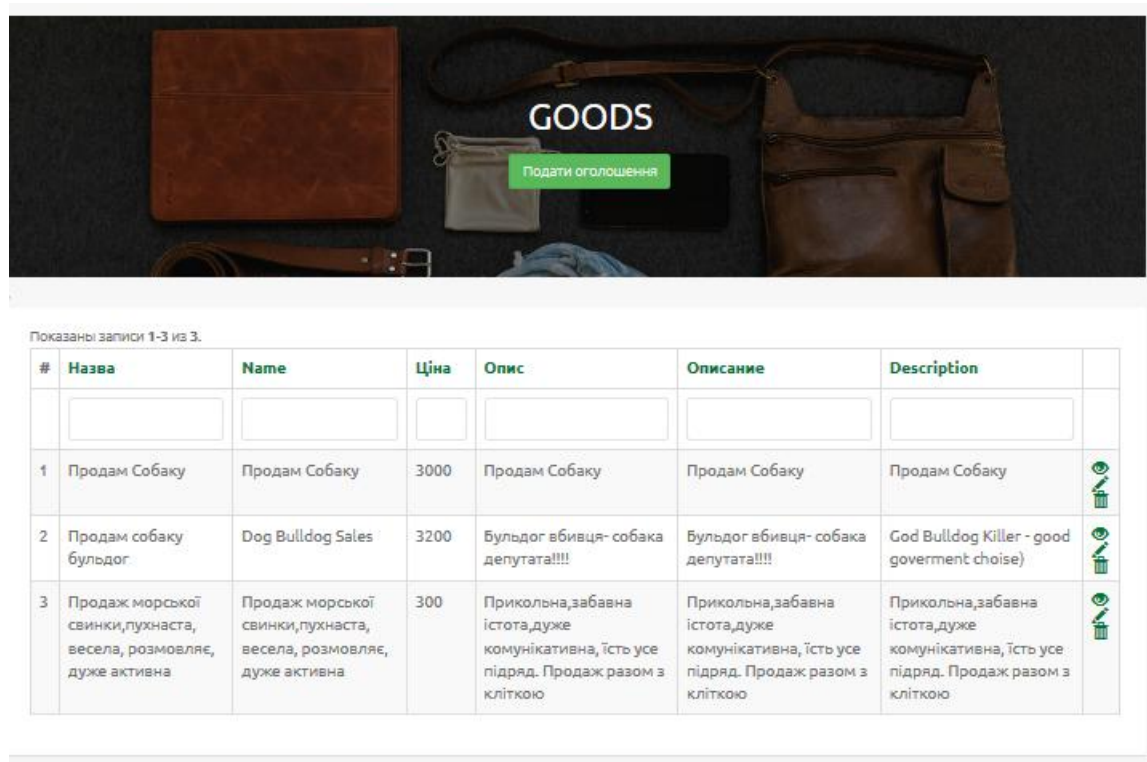


Рисунок 5.8 – Відображення оголошення у списку власних

Ціна

Стара ціна

Пошта

Необходимо заполнить «Пошта».

Номер телефону

Опис

Описание

Description

Зображення

[Огляд...](#) сніжок.jpg

[Редагувати](#)

Рисунок 5.9 – Застосування підказує користувачу, що не всі поля заповнені

8	Продам комп'ютер G-POWER	Продам комп'ютер G-POWER	5500	Продам комп'ютер G-POWER Характеристики: Використовувався півроку, у хорошому стані. Ціна: 5 500	Продам комп'ютер G-POWER Характеристики: Використовувався півроку, у хорошому стані. Ціна: 5 500	My_Dear_Girl	Computers	
9	Продам шиншиля	Продам шиншиля	600	Продам шиншиля	Продам шиншиля	volod	Animals	
10	Продаж морської свинки, пухнаста, весела, розмовляє, дуже активна	Продаж морської свинки, пухнаста, весела, розмовляє, дуже активна	300	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою	volodumur	Animals	

Рисунок 5.10– Відображення оголошення у списку усіх оголошень

Якщо користувач хоче переглянути деталі оголошення, необхідно натиснути кнопку Переглянути (рисунок 5.11).

10	Продаж морської свинки, пухнаста, весела, розмовляє, дуже активна	300	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою	volodumur	Animals	
----	---	-----	--	--	-----------	---------	--

Рисунок 5.11 – Віджет Переглянути

Деталі оголошення відображаються користувачу, як зареєстрованому, так і ні (рисунок 5.12).

Название	Продаж морської свинки, пухнаста, весела, розмовляє, дуже активна
Name	Продаж морської свинки, пухнаста, весела, розмовляє, дуже активна
Ціна	300
Стара ціна	(не задано)
Пошта	vsevolodnestaiko@gmail.com
Номер телефону	+380967355233
Опис	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Описание	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Description	Прикольна, забавна істота, дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
ID Категорії	1
Рейтинг товару	5
Товар відображається	yes
User	user.username
Category	category.name_en
Зображення	

Рисунок 5.12 – Перегляд оголошення

Щоб редагувати власне оголошення користувач повинен перейти у Мій Асортимент та натиснути Редагувати (рисунок 5.13, рисунок 5.14).

<div>Редагувати</div> <div>Видалити</div>	
ID	62
Назва	
Название	Продаж морської свинки,пухнаста, весела, розмовляє, дуже активна
Name	Продаж морської свинки,пухнаста, весела, розмовляє, дуже активна
Ціна	300
Стара ціна	(не задано)
Пошта	vsevolodnestaiko@gmail.com
Номер телефону	+380967355233
Опис	Прикольна,забавна істота,дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Описание	Прикольна,забавна істота,дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Description	Прикольна,забавна істота,дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
ID Категорії	1
Рейтинг товару	5
Товар відображається	yes

Рисунок 5.13 - Перегляд інформації, можливість редагування оголошення користувачем

Name	Продаж морської свинки,пухнаста, весела, розмовляє, дуже активна
Рейтинг товару	5
Товар відображається	yes
Ціна	300
Стара ціна	
Пошта	vsevolodnestaiko@gmail.com
Номер телефону	+380967355233
Опис	Прикольна,забавна істота,дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Описание	Прикольна,забавна істота,дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Description	Прикольна,забавна істота,дуже комунікативна, їсть усе підряд. Продаж разом з кліткою
Зображення	<div>Огляд...</div> <div>Файл не вибрано.</div>
<div>Редагувати</div>	

Рисунок 5.13 – Форма редагування оголошення

Щоб мати можливість керувати застосуванням було розроблено модуль адміністратора. Перейти до нього можна ввівши на сторінці входу наступні авторизаційні дані (рисунок 5.14):

Рисунок 5.14 – Дані для входу на сторінку адміністратора

Переходимо в меню адміністратора. Воно містить у собі наступні компоненти : категорії, асортимент та користувачі. У адміністратора є повноваження редагувати, переглядати та видаляти оголошення від усіх користувачів(рисунок5.15).

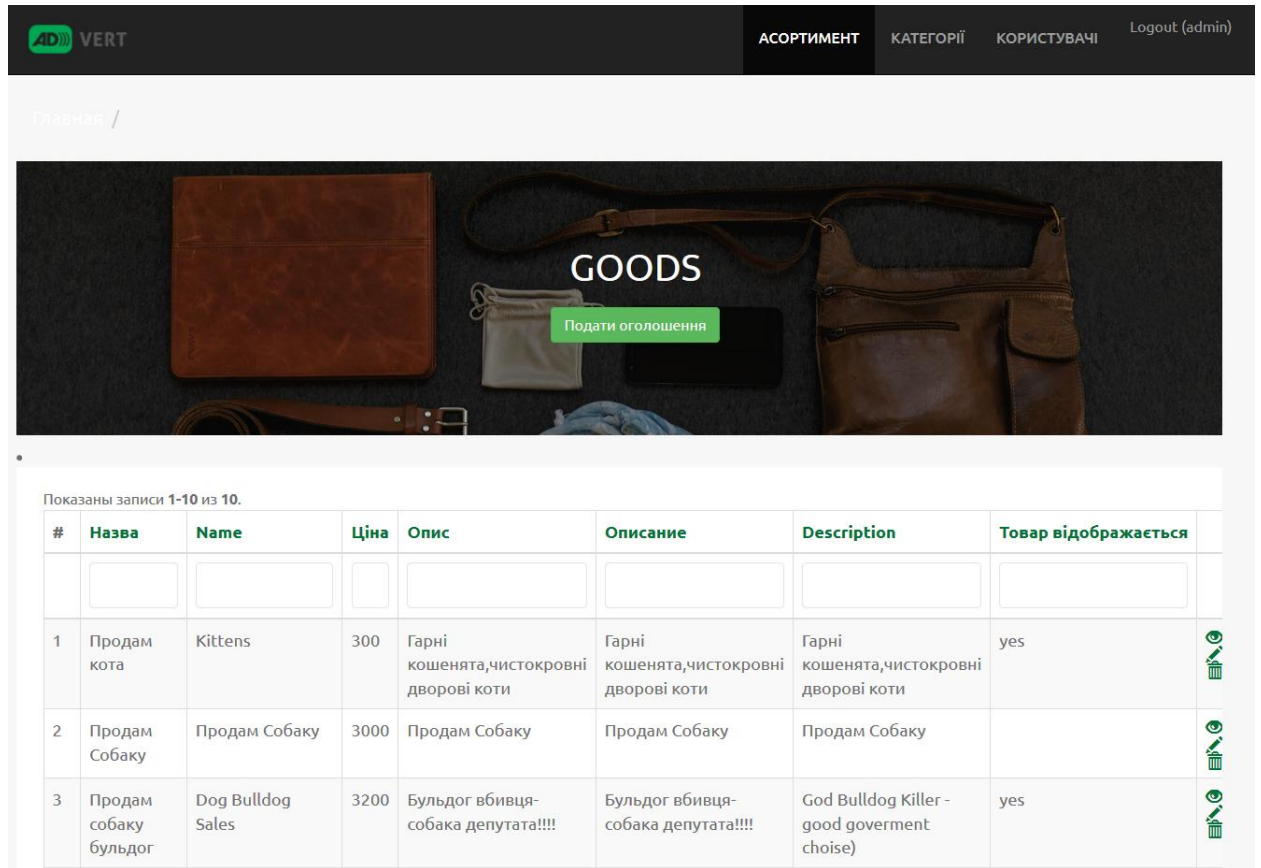


Рисунок 5.15 – Дані для входу на сторінку адміністратора

Натиснувши на кнопку Користувачі можна перейти на сторінку керування користувачами (рисунок 5.16). А також можна переглядати інформацію про користувача (рисунок 5.17). При видаленні користувача, обов'язково з'явиться відповідне повідомлення з підтвердженням дії (рисунок 5.18)

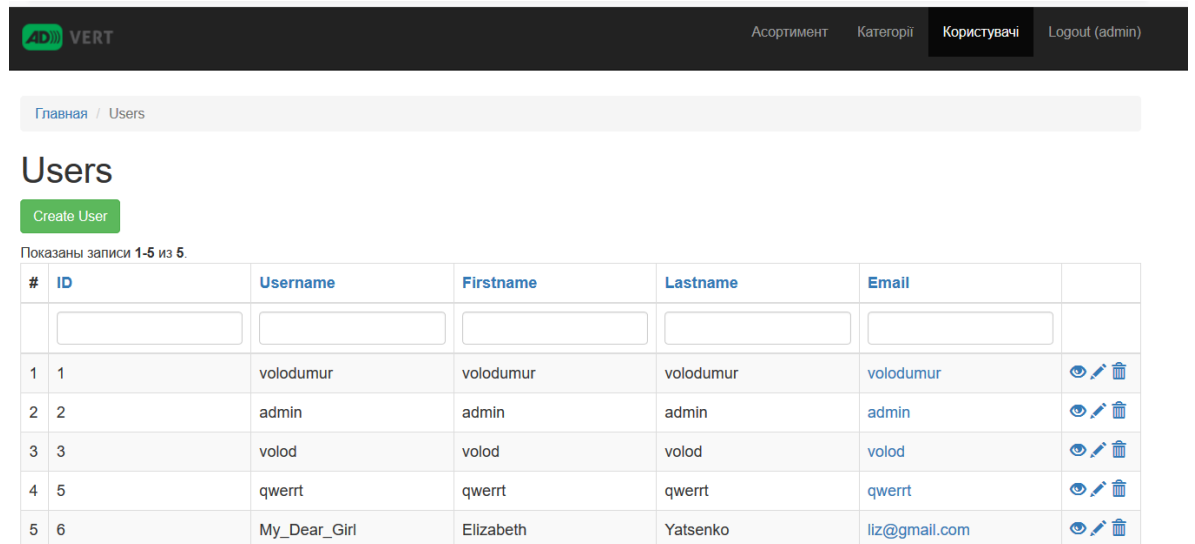


Рисунок 5.16 – Сторінка керування користувачами

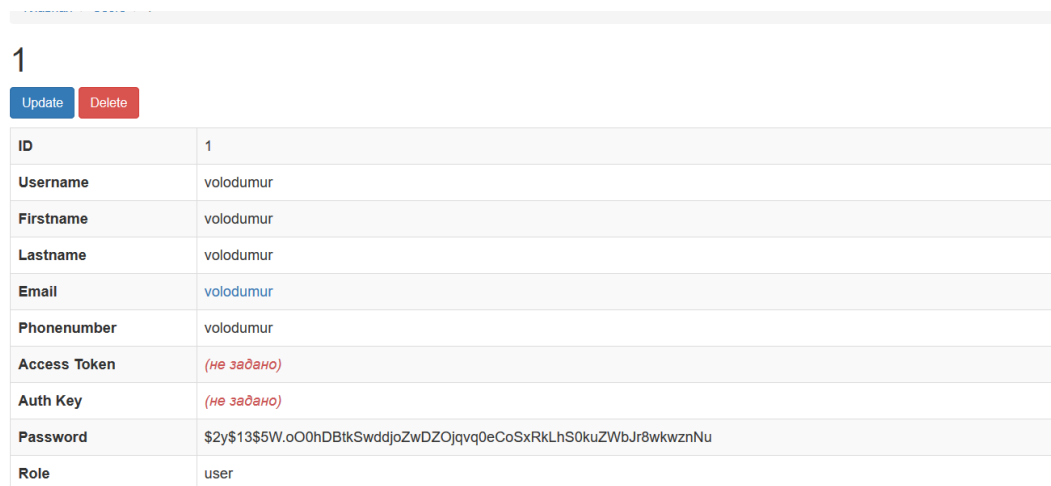


Рисунок 5.17 – Сторінка перегляду інформації про користувача

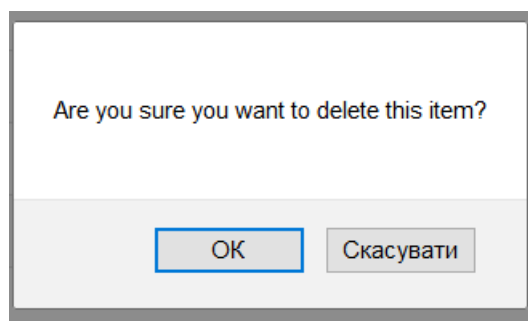


Рисунок 5.18 – Повідомлення про підтвердження видалення користувача

5.2 Мета випробувань

Метою випробувань являється перевірка відповідності функцій комплексу задач для дошки оголошень вимогам технічного завдання.

5.3 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.4 Результати випробувань

В процесі тестування були перевірена уся функціональність комплексу задач. У перелік випробувань інформаційної системи наведений у таблицях 5.1- 5.18.

Таблиця 5.1 – Введення логіну та паролю

Мета тесту:	Перевірка функції «Вхід в частину адміністрування»
Початковий стан КЗ	Відкрита «Вхід» сторінка сайту
Вхідні данні:	Логін та пароль
Схема проведення тесту:	Ввести у поле «Логін» логін, а у поле «Пароль»
	Ввести пароль, натиснути кнопку «Вхід»
Очікуваний результат:	Відкрита сторінка «Особистий кабінет»
Стан КЗ після проведення випробувань:	Відкрита сторінка «Особистий кабінет»

Таблиця 5.2 – Перевірка заповнення усіх атрибутів

Мета тесту:	Перевірка функції «Перевірка заповнення усіх атрибутів»
Початковий стан КЗ	Відкрита «Вхід» сторінка сайту
Вхідні данні:	Логін або пароль

Продовження таблиці 5.2

Схема проведення тесту:	Ввести у поле «Логін» логін, або у поле «Пароль» ввести пароль адміністратора, натиснути кнопку «Вхід»
Очікуваний результат:	Вікно повідомлення про помилку вводу. Вхід на сторінку «Особистий кабінет» не виконано «Неверное имя или пароль»
Стан КЗ після проведення випробувань:	Відкрита сторінка аутентифікації

Таблиця 5.3 – Перевірка правильності введених даних

Мета тесту:	Перевірка функції «Перевірка правильності введених даних»
Початковий стан КЗ	Відкрита «Вхід» сторінка сайту
Вхідні данні:	Логін та пароль
Схема проведення тесту:	Ввести у поле «Логін» будь-який логін, а у поле «Пароль» ввести будь-який пароль, натиснути кнопку «Вхід»
Очікуваний результат:	Вікно повідомлення про помилку вводу. Вхід на сторінку «Особистий кабінет» не виконано «Неверное имя или пароль»
Стан КЗ після проведення випробувань:	Відкрита сторінка аутентифікації

Таблиця 5.4 – Повернення на головну сторінку сайту

Мета тесту :	Перевірка функції «Перехід на головну сторінку сайту»
Початковий стан КЗ:	Відкрита сторінка аутентифікації
Вхідні дані:	

Продовження таблиці 5.4

Схема проведення тесту:	Натиснути кнопку «Повернутися на головну»
Очікуваний результат:	Відкрита головна сторінка сайту
Стан КЗ після проведення випробування:	Відкрита головна сторінка сайту

Таблиця 5.5 – Додавання оголошення

Мета тесту:	Перевірка функції «Додавання оголошення»
Початковий стан КЗ:	Відкрита сторінка «Додати оголошення»
Вхідні дані:	Новина та підпис
Схема проведення тесту:	Натиснути кнопку «Додати», ввести у полі «Оголошення» назву, у полі «Категорія» - обрати категорію, у полі «Опис» - додати опис товару, у полі «Фото» - додати зображення товару, натиснути кнопку «Додати»
Мета тесту:	Перевірка функції «Додавання оголошення»
Очікуваний результат:	На сторінці «Оголошення» з'явилося нове оголошення
Стан КЗ після проведення випробування:	Відкрита сторінка «Особистий кабінет»

Таблиця 5.6 – Редагування оголошення

Мета тесту:	Перевірка функції «Редагування видалення»
Початковий стан КЗ:	Відкрита сторінка «Особистий кабінет»
Вхідні дані:	

Продовження таблиці 5.6

Схема проведення тесту:	Натиснути кнопку «Редагувати» біля оголошення, яке потрібно відредагувати. У полі оголошення відредагувати оголошення, у полі інформація відредагувати інформацію. Натиснути кнопку
	«Застосувати»
Очікуваний результат:	Оголошення відредаговане
Стан КЗ після проведення випробування:	Відкрита сторінка «Особистий кабінет»

Таблиця 5.7 – Видалення оголошення

Мета тесту:	Перевірка функції «Видалення оголошення»
Початковий стан КЗ:	Відкрита сторінка «Особистий кабінет»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку «Видалити» біля оголошення, яке потрібно видалити. Підтвердити бажання видалити новину натиснувши кнопку «ОК»
Очікуваний результат:	Оголошення видалене
Стан КЗ після проведення випробування:	Відкрита сторінка «Особистий кабінет»

Таблиця 5.8 – Додавання фото при створенні оголошення

Мета тесту:	Перевірка функції «Додавання фото при створенні оголошення»
Початковий стан КЗ:	Відкрита сторінка «Додати оголошення»
Вхідні дані:	Файл фото, назва фото
Схема проведення тесту:	Натиснути кнопку «Добавити», заповнити усі поля на сторінці, натиснути кнопку «Добавити»

Продовження таблиці 5.8

Очікуваний результат:	Фото додано. Повідомленні про успіх додавання та фото відображається на тій самій сторінці.
Стан КЗ після проведення випробування:	Відкрита сторінка «Додавання оголошення»

Таблиця 5.9 – Перевірка заповнення усіх полів на сторінці

Мета тесту:	Перевірка функції «Перевірка заповнення усіх полів на сторінці»
Початковий стан КЗ:	Відкрита сторінка «Додавання оголошення»
Вхідні дані:	Файл фото або назва фото
Схема проведення тесту:	Натиснути кнопку «Добавити», заповнити усі поля на сторінці, натиснути кнопку «Добавити»
Очікуваний результат:	Вікно повідомлення про помилку вводу. Додавання фото не виконано.
Стан КЗ після проведення випробування:	Відкрита сторінка «Додавання оголошення»

Таблиця 5.10 – Редагування зображення

Мета тесту:	Перевірка функції «Редагування фотографії»
Початковий стан КЗ:	Відкрита сторінка «Редагування оголошення»
Вхідні дані:	Марка, Модель, Колір, Рік, Об'єм двигуна, Тип кузова, Тип пального, Тип коробки передач, Описання, Ціна, Пошта, Номер Телефону, Назва категорії.
Схема проведення тесту:	Натиснути кнопку «Огляд». Обрати зображення. Натиснути кнопку «Редагувати».
Очікуваний результат:	Запис відредагований
Стан КЗ після проведення випробування:	Відкрита сторінка «Мій Асортимент»

Таблиця 5.11 – Перевірка функції подачі оголошення незареєстрованим користувачем.

Мета тесту:	Перевірка функції «Перевірка функції подачі оголошення незареєстрованим користувачем. »
Початковий стан КЗ:	Відкрита сторінка «Асортимент»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку «Створити оголошення»
Очікуваний результат:	Система просить користувача авторизуватись
Стан КЗ після проведення випробування:	Відкрита сторінка «Вхід»

Таблиця 5.12 – Перевірка унікальності імені користувача

Мета тесту:	Перевірка функції «Перевірка унікальності імені користувача»
Початковий стан КЗ:	Відкрита сторінка «Реєстрація»
Вхідні дані:	Ім'я користувача, Ім'я, Прізвище, Пошта, Номер телефону.
Схема проведення тесту:	Заповнити усі поля на сторінці, натиснути кнопку «Зареєструватись»
Очікуваний результат:	Система виводить повідомлення: «Цей логін зайнятий іншим користувачем»
Стан КЗ після проведення випробування:	Відкрита сторінка «Реєстрація»

Таблиця 5.13 – Перевірка можливості редагування усіх оголошень адміністратором.

Мета тесту:	Перевірка функції «Перевірка можливості редагування усіх оголошень адміністратором»
Початковий стан КЗ:	Відкрита сторінка «Асортимент»
Вхідні дані:	Оголошення від усіх користувачів
Схема проведення тесту:	Натиснути кнопку «Редагувати», змінити усі, або окремі поля, натиснути кнопку «Редагувати»
Очікуваний результат:	Відкрита сторінка «Асортимент». Зміни видні як адміністратору, так і усім користувачам.
Стан КЗ після проведення випробування:	Відкрита сторінка «Асортимент»


Таблиця 5.14 – Додавання фото при редагуванні

Мета тесту:	Перевірка функції «Редагування фото»
Початковий стан КЗ:	Відкрита сторінка «Редагування оголошень»
Вхідні дані:	Файл фото, назва
Схема проведення тесту:	Натиснути кнопку «Редагувати», заповнити усі поля на сторінці, натиснути кнопку «Редагувати»
Очікуваний результат:	Додана нова фотографія
Стан КЗ після проведення випробування:	Відкрита сторінка «Мій Асортимент»

Таблиця 5.15 – Видалення фото при редагуванні

Мета тесту:	Перевірка функції «Редагування фото»
Початковий стан КЗ:	Відкрита сторінка «Редагування оголошень»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку «Видалити» біля фото, яке потрібно видалити. Підтвердити бажання видалити авто натиснувши кнопку «ОК»
Очікуваний результат:	Фото видалено
Стан КЗ після проведення випробування:	Відкрита сторінка «Редагування оголошень»


Таблиця 5.16 – Перегляд оголошення

Мета тесту:	Перевірка функції «Перегляд оголошення»
Початковий стан КЗ:	Відкрита сторінка «Асортимент»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку 
Очікуваний результат:	Відкрита сторінка «Перегляд оголошення»

Продовження таблиці 5.16

Стан КЗ після проведення випробування:	Відкрита сторінка «Перегляд оголошення»
--	---

Таблиця 5.17 – Перегляд категорій

Мета тесту:	Перевірка функції «Перегляд категорій»
Початковий стан КЗ:	Відкрита сторінка «Категорії»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку 
Очікуваний результат:	Відкрита сторінка «Перегляд категорій»
Стан КЗ після проведення випробування:	Відкрита сторінка «Перегляд категорій»

Таблиця 5.18 – Додавання категорії

Мета тесту:	Перевірка функції «Додавання категорії»
Початковий стан КЗ:	Відкрита сторінка «Створити категорію»
Вхідні дані:	Назва категорії та батьківська категорія
Схема проведення тесту:	Натиснути кнопку «Створити», ввести у полі «Назва» назву, у полі «Батьківська категорія» - обрати категорію, натиснути кнопку «Додати»
Очікуваний результат:	На сторінці «Категорія» з'явилась нова категорія
Стан КЗ після проведення випробування:	Відкрита сторінка «Категорії»

Висновок до розділу

У цьому розділі наведено керівництво користувача та адміністратора. Детально описана робота програми, а також наведено усі тести, які пройшов продукт, і які можна було тільки придумати.

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНІ ВИСНОВКИ

У ході виконання дипломного проекту була детально розглянуті питання, які виникають в процесі створення web-додатків, які вимагають поглиблених знань ООП, патерну MVC. Були виділені основні ключові етапи, притаманні процесу, та взаємозв'язки між ними.

Був проведений ґрунтовний аналіз предметного середовища, ретельно описаний бізнес-процес розробки.

На основі даних, отриманих в процесі аналізу, було сформульовано відповідну математичну задачу. Пропозиції користувачу тих товарів, які б для його, на думку статистики, підійдуть найкраще. Для розробки програмного забезпечення була використана мова php та фреймворк Yii2 basic.

Розроблена модель бази даних, яка дає змогу ефективно та надійно здійснювати доступ до даних, що надходять та використовуються в процесі створення плану виготовлення продукції на підприємстві та його прогнозуванні на майбутні періоди. Для управління базою даних обрана СУБД PhpMyAdmin MySQL.

Наведена детальна інструкція користувача по експлуатації комплексу задач, описана методика проведення випробувань, яка показує можливість введення програми в експлуатацію.

ПЕРЕЛІК ПОСИЛАНЬ

1. <https://uk.wikipedia.org/wiki/%D0%9E%D0%B3%D0%BE%D0%BB%D0%BE%D1%88%D0%B5%D0%BD%D0%BD%D1%8F>
2. http://ua.kulikoff.com.ua/desc_ob/
3. <https://onesky.tech/?p=231>
4. <https://www.olx.ua/uk/>
5. https://www.ebay.com/?ef_id=Cj0KCQjwsZ3kBRCnARIsAIuAV_SCxfBS
6. <https://www.amazon.com/>
7. <https://uk.wikipedia.org/wiki/EBay>
8. http://ela.kpi.ua/bitstream/123456789/26158/1/Kasianchyk_magistr.pdf
9. <http://ukrarticles.pp.ua/pk-internet/6236-kratkoe-opisanie-yazyka-php.html>
10. <https://uk.wikipedia.org/wiki/MySQL>
11. <https://uk.wikipedia.org/wiki/HTML>
12. <https://uk.wikipedia.org/wiki/CSS>
13. <https://www.yiiframework.com/doc/guide/2.0/en/intro-yii>

Додаток А
Тексти програмного коду

Тексти програмного коду
Інформаційна система підтримки процесу подачі оголошень
(Найменування програми(документа))

DVD-R

(Вид носія даних)

10 арк. 50 Мб

(Обсяг програми(документа), арк.,) Мб)

Клас User :

```
namespace app\models;
```

```
use Yii;
```

```
use yii\base\NotSupportedException;
```

```
use yii\db\ActiveRecord;
```

```
use yii\helpers\Security;
```

```
use yii\web\IdentityInterface;
```

```
/**
```

```
 * This is the model class for table "user".
```

```
 *
```

```
 * @property string $id
```

```
 * @property string $username
```

```
 * @property string $firstname
```

```
 * @property string $lastname
```

```
 * @property string $email
```

```
 * @property string $phonenummer
```

```
 * @property string $access_token
```

```
 * @property string $auth_key
```

```
 * @property string $password
```

```
 * @property string $role
```

```
 * @property string $sid
```

```
 * @property string $date_expire
```

```
 * @property Assortiment[] $assortiments
```

```
 */
```

```
class User extends \yii\db\ActiveRecord implements \yii\web\IdentityInterface
```

```
{
```

```
    /**
```

```
     * {@inheritdoc}
```

```
    */
```

```
    public static function tableName()
```

```
    {
```

```
        return 'user';
```

```
    }
```

```
    public static function findIdentity($id)
```

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

```
{
    return static::findOne($id);
}

/**
 * {@inheritdoc}
 */

public static function findIdentityByAccessToken($token, $type = null)
{
    foreach (self::$users as $user) {
        if ($user['accessToken'] === $token) {
            return new static($user);
        }
    }

    return null;
}

/**
 * {@inheritdoc}
 */

/*
public static function checkUserLogin($sid, $hash){
    $user = static::findOne(['sid' => $sid]);
    if(md5($user['id']) === $hash)
        return false;
    return false;
}

public static function Login($login, $password){
    $user = static::findOne(['username' => $login, 'password' => \Yii::$app->security->generatePasswordHash($password)]);
    if($user)
        Yii::$app->user->login($user);
    if(date('d-m-Y') <= $user['date_expires']) {
        $sid = $user['sid'];
    } else {
        $sid = md5(rand(0,1000));
    }
    $user['sid'] = $sid;
```

```
        return ['user' => $user,
                'hash' => $hash]; //[$sid, hash($user['id'])];
    }

    */

/**
 * Finds user by username
 *
 * @param string $username
 * @return static|null
 */
public static function findByUsername($username)
{
    return static::findOne(['username' => $username]);
}

/**
 * {@inheritdoc}
 */
public function getId()
{
    return $this->id;
}

/**
 * {@inheritdoc}
 */
public function getAuthKey()
{
    return null;
}

/**
 * {@inheritdoc}
 */
public function validateAuthKey($authKey)
{
    return $this->authKey === $authKey;
}
```

```

/**
 * Validates password
 *
 * @param string $password password to validate
 * @return bool if password provided is valid for current user
 */
public function validatePassword($password)
{
    return \Yii::$app->security->validatePassword($password, $this->password);
}

/**
 * {@inheritdoc}
 */
public function rules()
{
    return [
        [['username', 'firstname', 'lastname', 'email', 'phonenumber', 'password'], 'required'],
        [['username', 'firstname', 'lastname', 'email', 'phonenumber', 'access_token', 'auth_key', 'password',
'role'], 'string', 'max' => 255],
        [['username'], 'unique'],
    ];
}

/**
 * {@inheritdoc}
 */
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'username' => 'Username',
        'firstname' => 'Firstname',
        'lastname' => 'Lastname',
        'email' => 'Email',
        'phonenumber' => 'Phonenumber',
        'access_token' => 'Access Token',
        'auth_key' => 'Auth Key',
        'password' => 'Password',
        'role' => 'Role',
    ];
}

```



```
'sid' => 'sid',
];
}

/**
 * @return \yii\db\ActiveQuery
 */
public function getAssortiments()
{
    return $this->hasMany(Assortiment::className(), ['user_id' => 'id']);
}
}

Клас LoginForm :
<?php

namespace app\models;

use Yii;
use yii\base\Model;

/**
 * LoginForm is the model behind the login form.
 *
 * @property User|null $user This property is read-only.
 */
class LoginForm extends Model
{
    public $username;
    public $password;
    public $rememberMe = true;

    private $_user = false;

    /**
     * @return array the validation rules.
     */
    public function rules()
    {
        return [
```

```

        // username and password are both required
        [['username', 'password'], 'required'],
        // rememberMe must be a boolean value
        ['rememberMe', 'boolean'],
        // password is validated by validatePassword()
        ['password', 'validatePassword'],
    ];
}
public function attributeLabels()
{
    return [
        'username' => Yii::t('common', 'Логін'),
        'password' => Yii::t('common', 'Пароль'),
        'rememberMe' => Yii::t('common', 'Пам'ятати мене'),
    ];
}

/**
 * Validates the password.
 * This method serves as the inline validation for password.
 *
 * @param string $attribute the attribute currently being validated
 * @param array $params the additional name-value pairs given in the rule
 */
public function validatePassword($attribute, $params)
{
    if (!$this->hasErrors()) {
        $user = $this->getUser();

        if (!$user || !$user->validatePassword($this->password)) {
            $this->addError($attribute, 'Incorrect username or password.');
```

```

        }
    }
}

/**
 * Logs in a user using the provided username and password.
 * @return bool whether the user is logged in successfully
 */
public function login()
{

```

```

        if ($this->validate()) {
            return Yii::$app->user->login($this->getUser(), $this->rememberMe ? 3600*24*30 : 0);
        }
        return false;
    }

```

```
/**
```

```
 * Finds user by [[username]]
```

```
 *
```

```
 * @return User|null
```

```
 */
```

```
public function getUser()
```

```
{
```

```
    if ($this->_user === false) {
```

```
        $this->_user = User::findByUsername($this->username);
```

```
    }
```

```
        return $this->_user;
```

```
    }
```

```
}
```

Клас Signup :

```
<?php
```

```
namespace app\models;
```

```
use Yii;
```

```
use yii\base\Model;
```

```
use yii\db\ActiveRecord;
```

```
class SignupForm extends Model
```

```
{
```

```
    public $username;
```

```
    public $firstname;
```

```
    public $lastname;
```

```
    public $email;
```

```
    public $phonenum;
```

```
    public $password;
```

```
    public function rules()
```

```
{
```

```

return [
    [['username','firstname','lastname','email','phonenumber', 'password'], 'required', 'message' =>
'Заповніть поле'],
    ['username','unique','targetClass' => User::className(), 'message' => 'Цей логін зайнятий іншим
користувачем'],
];
}

```

```

public function attributeLabels()

```

```

{
    return [
        'username' => Yii::t('common','Логін'),
        'firstname' => Yii::t('common','Імя'),
        'lastname' => Yii::t('common','Прізвище'),
        'email' =>    Yii::t('common','Пошта'),
        'phonenumber' => Yii::t('common','Номер телефону'),
        'password' =>  Yii::t('common','Пароль'),
    ];
}

```

```

}

```

Клас Assortiment :

```

<?php

```

```

namespace app\models;

```

```

use Yii;

```

```

//use app\models\AdvertImage;

```

```

use app\models\Category;

```

```

use yii\db\ActiveRecord;

```

```

use rico\yii2images\models\Module;

```

```

use rico\yii2images\models\Image;

```

```

/**

```

```

 * This is the model class for table "{{%tbl_assortiment}}".

```

```

 *

```

```

 * @property integer $id

```

```

 * @property string $description_ru

```

```

 * @property string $description_et

```

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

* @property string $description_en
* @property integer $category_id
* @property integer $rating
* @property string $shown
* @property string $name_ru
* @property string $name_et
* @property string $name_en
* @property double $prise
* @property double $prise_old
*
* @property TblCategory $category
*/

class Assortiment extends \yii\db\ActiveRecord
{
    const STATUS_DELETED = 0;
    const STATUS_ACTIVE = 10;
    /**
     * @inheritdoc
     */
    // assortment_img value
    public $image;
    public $gallery;

    public function behaviors()
    {
        return [
            'image' => [
                'class' => 'rico\yii2images\behaviors\ImageBehave',
            ]
        ];
    }
    public static function tableName()
    {
        return '{{%assortiment}}';
    }

    public function getUser() {
        return $this->hasOne('app\models\User', ['id' => 'user_id']);
    }

```

```

/* public function getImages() {
    return $this->hasMany('app\models\AdvertImage', ['assortment_id' => 'id']);
}*/

public function getCategory()
{
    return $this->hasOne(Category::className(), ['id' => 'category_id']);
}

/**
 * @inheritdoc
 */
public function rules()
{
    return [
        [['description_ru', 'description_et', 'description_en', 'category_id', 'user_id', 'tag', 'rating',
        'prise', 'email'], 'required'],
        [['description_ru', 'description_et', 'description_en', 'shown'], 'string'],
        [['category_id', 'user_id', 'rating'], 'integer'],
        [['prise', 'prise_old'], 'number'],
        [['name_ru', 'name_et', 'name_en', 'email', 'phone', 'tag'], 'string', 'max' => 255],
        [['image'], 'file', 'extensions' => 'png, jpg'],
        [['gallery'], 'file', 'extensions' => 'png, jpg', 'maxFiles' => 4]
    ];
}

/**
 * @inheritdoc
 */
public function attributeLabels()
{
    return [
        'id' => Yii::t('app', 'ID'),

```

```

'description_ru' => Yii::t('common', 'Опис'),
'description_et' => Yii::t('common', 'Описание'),
'description_en' => Yii::t('common', 'Description'),
'category_id' => Yii::t('common', 'ID Категорії'),
// 'user_id' => Yii::t('common', 'ID Користувача'),
'rating' => Yii::t('common', 'Рейтинг товару'),
'shown' => Yii::t('common', 'Товар відображається'),
'name_ru' => Yii::t('common', 'Назва'),
'name_et' => Yii::t('common', 'Название'),
'name_en' => Yii::t('common', 'Name'),
'email' => Yii::t('common', 'Пошта'),
'phone' => Yii::t('common', 'Номер телефону'),
'prise' => Yii::t('common', 'Ціна'),
'prise_old' => Yii::t('common', 'Стара ціна'),
'image' => Yii::t('common', 'Зображення')

];
}

public function upload(){

    if ($this->validate()){
        $path = 'uploads/store/'.$this->image->baseName. '.'.$this->image->extension;
        $this->image->saveAs($path);
        $this->attachImage($path);
        return true;
    }else{
        return false;
    }

}

/**
 * @return \yii\db\ActiveQuery
 */

}Клас AssortimentMine
<?php

namespace app\models;
```

```

use Yii;

//use app\models\AdvertImage;
use app\models\Category;
use yii\db\ActiveRecord;
use rico\yii2images\models\Module;
use rico\yii2images\models\Image;

/**
 * This is the model class for table "{{tbl_Assortimentmine}}".
 *
 * @property integer $id
 * @property string $description_ru
 * @property string $description_et
 * @property string $description_en
 * @property integer $category_id
 * @property integer $rating
 * @property string $shown
 * @property string $name_ru
 * @property string $name_et
 * @property string $name_en
 * @property double $prise
 * @property double $prise_old
 *
 * @property TblCategory $category
 */
class Assortimentmine extends \yii\db\ActiveRecord
{
    const STATUS_DELETED = 0;
    const STATUS_ACTIVE = 10;

    /**
     * @inheritdoc
     */

    // Assortimentmine_img value
    public $image;
    public $gallery;

    public function behaviors()
    {
        return [

```



```

        'image' => [
            'class' => 'rico\yii2images\behaviors\ImageBehave',
        ]
    ];
}

public static function tableName()
{
    return '{{%Assortiment}}';
}

public function getUser() {
    return $this->hasOne('app\models\User', ['id' => 'user_id']);
}

/* public function getImages() {
    return $this->hasMany('app\models\AdvertImage', ['Assortimentmine_id' => 'id']);
} */

public function getCategory()
{
    return $this->hasOne(Category::className(), ['id' => 'category_id']);
}

/**
 * @inheritdoc
 */
public function rules()
{
    return [
        [['description_ru', 'description_et', 'description_en', 'category_id','user_id', 'rating', 'prise','email'],
        'required'],
        [['description_ru', 'description_et', 'description_en', 'shown'], 'string'],
        [['category_id','user_id', 'rating'], 'integer'],
        [['prise', 'prise_old'], 'number'],
        [['name_ru', 'name_et', 'name_en','email','phone'], 'string', 'max' => 255],
        [['image'], 'file', 'extensions' => 'png, jpg'],
        [['gallery'], 'file', 'extensions' => 'png, jpg', 'maxFiles' => 4]
    ];
}

```

```

/**
 * @inheritdoc
 */
public function attributeLabels()
{
    return [
        'id' => Yii::t('app', 'ID'),
        'description_ru' => Yii::t('common', 'Опис'),
        'description_et' => Yii::t('common', 'Описание'),
        'description_en' => Yii::t('common', 'Description'),
        'category_id' => Yii::t('common', 'ID Категорії'),
        'user_id' => Yii::t('common', 'ID Користувача'),
        'rating' => Yii::t('common', 'Рейтинг товару'),
        'shown' => Yii::t('common', 'Товар відображається'),
        'name_ru' => Yii::t('common', 'Назва'),
        'name_et' => Yii::t('common', 'Название'),
        'name_en' => Yii::t('common', 'Name'),
        'email' => Yii::t('common', 'Пошта'),
        'phone' => Yii::t('common', 'Номер телефону'),
        'prise' => Yii::t('common', 'Ціна'),
        'prise_old' => Yii::t('common', 'Стара ціна'),
        'image' => Yii::t('common', 'Зображення')
    ];
}

public function upload(){
    if ($this->validate()){
        $path = 'uploads/store/'.$this->image->baseName. '.'.$this->image->extension;
        $this->image->saveAs($path);
        $this->attachImage($path);
        return true;
    }else{
        return false;
    }
}

/**
 * @return \yii\db\ActiveQuery

```

ДП ІС-5223.1181-с.ПЗ

*/

					ДП ІС-5223. 1181-с.ПЗ	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”

Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проекту

_____ М. І. Цюцюра
(підпис) (ініціали, прізвище)

“16” квітня 2019 р.

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“17” квітня 2019 р.

Інформаційна система підтримки процесу подачі оголошень

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр ДП ІС-5223.1181-с.ТЗ

На 11 сторінках

Київ – 2019 року

ЗМІСТ

1	ЗАГАЛЬНА ІНФОРМАЦІЯ.....	3
	1.1. Мета розробки сайту.....	3
	1.2 Завдання сайту.....	3
2	ВИМОГИ ДО САЙТУ.....	4
	2.1 Основні вимоги.....	4
	2.2. Зовнішня частина сайту.....	4
	2.3 Внутрішня частина сайту.....	6
	2.3.1 Система користувачів.....	6
	2.4 Вимоги до технічної частини сайту.....	7
	2.5 Вимоги до дизайну.....	9
3	ПОРЯДОК ВИКОНАННЯ РОБІТ.....	11

					ДП ІС-5223.1181-с.ПЗ			
		Прізвище	Підпис	Дата	Інформаційна система підтримки процесу подачі оголошень	Літ.	Арк.	Аркушів
Розроб.	Уманський В.А.							
Перевірів.	Цюцюра М.І.						2	11
						КПІ ім.І.Сікорського ФІОТ кафедра АСОІУ гр. ІС-52		
Н. кон.	Халус О.А.							
Затв.	Павлов О.А.							

1 ЗАГАЛЬНА ІНФОРМАЦІЯ

1.1 Мета розробки сайту

- розроблення веб – додатку дошки оголошень орієнтованої на іноземних замовників(роботодавців, продавців, продавців послуг);
- створення власного лого та стилю як основи для візуальної презентації та підвищення впізнаваності;
- формування та підтримка позитивного іміджу;
- поглиблення взаємодії з закордонними та національними роботодавцями.

1.2 Завдання сайту

- формування єдиного всесвітнього ресурсу з розміщення оголошень;
- забезпечення своєчасного розміщення інформації та підтримки процесу подачі оголошень;
- підвищення впізнаваності та ідентифікації шляхом використання власного логотипу та фірмового стилю;
- незалежний майданчик для розміщення оголошень;
- забезпечення ефективних двосторонніх комунікацій і каналів зворотного зв'язку.

2 ВИМОГИ ДО САЙТУ

2.1 Основні вимоги

Потрібно створити простий та інтуїтивно зрозумілий додаток, що повинен стати єдиною системою, що включає дві частини:

— зовнішня, що виконує презентаційні функції продукту та інформаційне забезпечення зовнішніх користувачів. Має бути єдиний підхід до керування вмістом сайту і можливість виконання базових операцій з його підтримки користувачам, що мають на це права, але не мають спеціальної підготовки в області програмування чи html-кодування, за винятком загальних навичок роботи з комп'ютером та веб-браузером. Керування контентним наповненням сайту -дошки оголошень здійснюється уповноваженими користувачами з особистої сторінки адміністратора.

— внутрішньої, що забезпечуватиме персоналізований вхід авторизованих користувачів та надаватиме їм повноваження або каревати власними оголошеннями, або права адміністратора.

2.2. Зовнішня частина сайту

2.2.1. Структура інтернет-сайту

Основне меню:

- головна;
- про нас;
- реєстрація;
- вхід;
- категорії;
- оголошення;
- контакти.

Службове меню:

- пошук;
- вибір мови (українська/англійська);
- авторизація користувача.

Розділ «Асортимент» містить усі оголошення користувачів. Також відбувається модерований відбір оголошень, дотичних до тематики імплементації Угоди про асоціацію від інформаційних агенцій.

Розділ «Про нас» містить біографічну та контактну інформацію щодо розробників сайту.

Розділ «Категорії» містить перелік категорій товарів, послуг та роботи, які можна розмістити на ресурсі. Перелік категорій постійно доповнюється та оновлюється.

Розділ «Користувачі» містить інформацію про усіх користувачів, доступний тільки для адміністратора сайту. Адміністратор має можливість переглядати та керувати користувачами.

Розділ «Місцезнаходження» містить інформацію про положення пропозиції, та полегшує пошук того чи іншого товару.

Розділ «Контакти» надає адресу (, контактні телефони та e-mail для зв'язку з службою підтримки сайту.

2.2.2. Мови сайту

Офіційними мовами є українська й англійська. Всі сторінки створюються двома офіційними мовами. Деяка інформація, за рішення адміністратора сайту, може подаватись лише англійською мовою.

2.2.3. Функціональна архітектура інтернет-сайту

До складу зовнішньої частини порталу входять наступні підсистеми:

- система сторінок;
- пошук;
- система оголошень;
- авторизація користувача;
- реєстрація;
- система категорій;
- система користувачів;
- система сторінок.

Більшість сторінок в структурі сайту складається із сторінок, які мають динамічний функціонал і містять лише контент із деякими картинками який редагується як із адміністративної частини, так і з користувацької авторизованими користувачами:

- пошук;
- система оголошень.

Містить інформацію, введену адміністратором та користувачем ресурсу. Може містити текстовий контент, картинки в форматі .jpeg та .png.

Авторизація користувача

Містить поля «Логін», «Пароль», «Написати листа адміністратору». Подібний доступ суворо модерується і надається лише зареєстрованим користувачам для можливості публікації власних оголошень у розділі «Асортимент». «Написати листа адміністратору» містить посилання на електронну пошту на яку користувач надсилає повідомлення з проханням надати доступ.

2.3 Внутрішня частина сайту

Внутрішня частина сайту забезпечуватиме персоналізований вхід користувачів на сайт та дозволить їм публікувати власні оголошення з відповідною модерацією.

2.3.1 Система користувачів

Для управління доступом користувачів до внутрішньої частини сайту та захисту впроваджено систему розмежування доступу (RBAC).

- Адміністратор має максимальні права доступу;
- гості – незареєстровані відвідувачі сайту, мають доступ лише до перегляду матеріалів у розділах «Асортимент» та «Категорії»;
- користувачі – зареєстровані відвідувачі сайту, які мають доступ до додавання матеріалів лише у розділ «Асортимент» та «Категорії».

Користувач володіє наступними характеристиками:

- прізвище та ім'я;
- логін;
- пароль;
- номер телефону;
- електронну пошту.

2.4 Вимоги до технічної частини сайту

2.4.1 Вимоги до архітектури порталу

Архітектура порталу повинна забезпечувати масштабованість і розширення системи. Додавання додаткового функціоналу повинне відбуватися за рахунок додавання додаткових модулів без суттєвої модернізації вже існуючих модулів. Архітектура порталу має передбачати незалежність модулю системи відображення інформації від модулю зберігання та керування інформацією. Зміна структури і дизайну сторінок не повинна викликати зміну звертань та запитів до БД порталу (MVC pattern). Архітектура порталу має передбачати незалежність реалізації системи від апаратної платформи і серверної операційної системи. Підтримку діяльності сайту має передбачати можливість виконання базових операцій з його підтримки людям, що не мають спеціальної підготовки в області програмування чи html-кодування, за винятком загальних навиків роботи з комп'ютером та веб-браузером.

2.4.2 Вимоги до програмної платформи та коду

- Сайт має бути розроблений з використанням серверу Hammp;
- кодування тексту, що має використовуватись на усіх сторінках усіх мовних версій та в БД – UTF-8;
- YiiFramework версії 2 basic;

- виконання функцій адміністратора сайту повинно бути доступним виключно за протоколом HTTPS (з криптографічним захистом обміну даними між сайтом та робочим місцем);
- програмне забезпечення сайту повинно забезпечувати функціонування сайту в режимі 24 години на добу, 7 діб на тиждень, 365 днів на рік (за умови безвідмовного функціонування апаратного забезпечення);
- програмне забезпечення має забезпечувати одночасну (паралельну) роботу з сайтом 1000 (тисяча) анонімних відвідувачів протягом 8 (восьми) годин безперервно, з паралельною обробкою та публікацією 500 (п'ятисот) інформаційних матеріалів в різних розділах сайту, без жодної відмови в обслуговуванні та без погіршення часу завантаження сторінки.

2.4.3 Резервне копіювання

Система повинна забезпечувати можливість резервного копіювання та відновлення через інтерфейс адміністратора.

2.4.4 Вимоги до надійності та безпеки

Програмні механізми порталу повинні забезпечувати функціонування в цілодобовому режимі з допустимими перервами на профілактику, з попереднім інформуванням користувачів.

Портал повинен бути захищений від MySQL-ін'єкцій, а також інших засобів втручання до безперервної роботи сайту з боку його програмної платформи.

2.4.5 Вимоги до формування адресного простору сайту

Формування адресного простору сайту має передбачати створення для кожного матеріалу сайту унікальної адреси (permalink). Це необхідно для зрозумілого користувачеві поточного положення в ієрархії сайту та коректної

індексації пошуковими системами, за допомогою дружніх URL-адрес (pretty URL's).

2.5 Вимоги до дизайну

Загальний дизайн сторінок сайту має базуватися на стабільних стандартах та не повинен використовувати можливостей, доступних тільки в окремих браузерях.

Логотип та фірмовий стиль розробляється для уніфікованого оформлення матеріалів, зокрема, сайту, та формування візуальної впізнаваності. Логотип та фірмовий стиль мають бути лаконічними, діловим та промовистим, який би вписувався в ідею створення та функціонування Advert Tech. Логотип та фірмовий стиль стане основою для подальшої розробки бренд-буку.

Графічні матеріали (фотографії та малюнки) створюються у форматах JPG (фотографії) та PNG (малюнки) з обов'язковою оптимізацією для відображення у Веб за розміром і кількістю кольорів. Графічні матеріали повинні коректно (без суттєвих спотворень) відображатись у Веб-безпечній палітрі: 256 кольорів. Дизайн порталу повинен відповідати усім сучасним нормам та напрямкам у веб-дизайні, бути простим й одночасно сучасним та інформативним.

Інтерфейс користувача повинен забезпечувати наглядне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації та швидкий і логічний перехід між розділами.

Навігаційні елементи повинні забезпечувати однозначне розуміння користувачами їх змісту: лінки на сторінки повинні містити заголовки, умовні позначення відповідати загальноприйнятим. Графічні елементи навігації повинні містити альтернативний підпис.

Всі лінки на власні сторінки порталу повинні відкриватися в цьому ж вікні браузера. У випадку якщо на порталі міститься лінк на сторонній сайт, він повинен відкриватися у новому вікні.

Мінімальна роздільна здатність під яку розробляється дизайн – 1024x768 пікселів (ширина зони основного контенту – не більше 980 пікселів), дизайн масштабується в межах від 980 до 1200 пікселів. При вищих роздільних здатностях екрану користувачів дизайн повинен відцентровуватись до середини, а з боків з'являться фонове зображення.

2.5.1 Зручність користування сайтом

- Використовувати прості випадаючі меню для представлення змісту розділів;

- об'єм сторінки не може перевищувати 100-120 Кб;

- час відкриття сторінки – максимум до 30 секунд за умови швидкості передачі даних на рівні 28 800 біт/с;

Сайт має функціонувати при використанні відвідувачами найбільш поширених браузерів :

- Microsoft Internet Explorer версій не нижче 7;

- Opera версій не нижче 10.0;

- Google Chrome версій не нижче 21.0;

- Mozilla Firefox версій не нижче 27.0;

- Safari.

3 ПОРЯДОК ВИКОНАННЯ РОБІТ.

Терміни

Проект передбачає реалізацію сайту в кілька етапів:

Етап I. Проектування

Узгодження технологій та рішень

Етап II. Розробка дизайну

Початок робіт над дизайном: створення логотипу та фірмового стилю.

Узгодження з замовником

Створення концепції дизайну сайту. Узгодження з замовником.

Етап III. Верстка сайту

- Верстка сторінок;
- front-end програмування;
- налаштування структури проекту;
- розробка функціоналу авторизації□;
- розробка функціоналу двомовності - українська та англійська мови;
- розробка функціоналу статей□;
- розробка функціоналу новин;
- розробка функціоналу коментарів;
- розробка функціоналу користувачів;
- розробка системи адміністрування.

Етап IV. Тестування та публікація сайту

Бажаний термін завершення роботи над сайтом та його публікація - 30 червня 2019 року.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проекту

(підпис) Цюцюра М.І.
(ініціали, прізвище)

“13” травня 2019 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

(підпис) О.А.Павлов
(ініціали, прізвище)

“14” травня 2019 р.

«Інформаційна система підтримки процесу подачі оголошень»

ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

Шифр ДП ІС-5223.1181-с.ПМВ

на 16 сторінках

Київ – 2019 року

ЗМІСТ

1 ОБ'ЄКТ ВИПРОБУВАНЬ	3
1.1 Найменування програми.....	3
1.2 Область застосування	3
1.3 Умовне позначення програми.....	3
2 МЕТА ВИПРОБУВАНЬ	4
3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ	5
3.1. Вимоги до функціональних характеристик.....	5
3.2. Вимоги до складу виконуваних функцій.....	5
4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ.....	6
5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ	7
6 РЕЗУЛЬТАТИ ВИПРОБУВАНЬ	8
6.1.Функціональне тестування.....	8
6.2.Модульне тестування.....	14

					ДП ІС-5223.1181-с.ПЗ		
		Прізвище	Підпис	Дата	Інформаційна система підтримки процесу подачі оголошень		
Розроб.	Уманський В.А.						
Перевірів.	Цюцюра М.І.						
Н. кон.	Халус О.А.						
Затв.	Павлов О.А.				<div>Літ.</div> <div>Арк.</div> <div>Аркушів</div> <div>2</div> <div>16</div> <div>КПІ ім.І.Сікорського ФІОТ кафедра АСОІУ гр. ІС-52</div> <div>2</div>		

1 ОБ'ЄКТ ВИПРОБУВАНЬ

1.1 Найменування програми

Об'єктом випробувань є інформаційна система AdvertTech, призначена для підтримки процесу подачі оголошень, що забезпечує розмежований контроль доступу за роллю користувачів. Система орієнтована на іноземних замовників(роботодавців, продавців, продавців послуг). Створено власне лого та стиль як основи для візуальної презентації та підвищення впізнаваності, і закладений у дизайн функціонал.

1.2 Область застосування

Система знаходить застосування у області торгівлі, пропозиції послуг та роботи. Це аналог популярного зараз в Україні сайту OLX, але з можливістю розширення аудиторії системи, та переходу на європейську тогрівлю, наприклад, можливість безпосередньо обрати та замовити будь-яку лізингову техніку, чи знайти вакансію за кордоном без участі сторонніх фірм-брокерів.

1.3 Умовне позначення програми

Надалі інформаційна система підтримки процесу подачі оголошень AdvertTech іменується як КЗ.

2 МЕТА ВИПРОБУВАНЬ

Метою випробувань являється перевірка відповідності функцій комплексу задач для дошки оголошень вимогам технічного завдання. Перевірка його надійності та стабільності у роботі.

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1. Вимоги до функціональних характеристик

Робота системи повинна бути стабільною, без перебоїв у роботі операційної системи. Функціонування програми не повинне призводити до перебою роботи операційної системи. Організація подачі інформації повинна передбачати захист від уведення некоректних даних та заповненості полів.

3.2. Вимоги до складу виконуваних функцій

Функції системи повинні працювати стабільно, виконувати свої задачі, незалежно від перевантаженості системи. Функції контролю доступу не повинні виходити з ладу, і надавати керування всім процесом адміністратору, а користувачам надавати доступ тільки до керування власними оголошеннями, а незареєстрованим користувачам тільки право перегляду загальної кількості оголошень. Функції, що відповідають за зміну мови повинні завжди коректно відпрацьовувати, для цього було створено словники як і в коді програми, так і резервну копію в базі даних, а також в кореневій папці проекту 'common'.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Випробування проводяться на основі наступних документів:

–ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;

–ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ

Програма працює відповідно до умов експлуатації на IBM сумісних ПК з ОС Windows XP/2003/Vista/Windows7/Windows10. А також з ОС Linux і

MAC OS. Система функціонує при використанні відвідувачами найбільш поширених браузерів:

Microsoft Internet Explorer версій не нижче 7/ Opera версій не нижче 10.0/ Google Chrome версій не нижче 21.0./ Mozilla Firefox версій не нижче 27.0/ Safari

Порядок проведення випробувань наступний:

Запуск програми здійснюється через браузер користувачем . Далі вибирається пункт реєстрація після чого заповнюється форма, видана програмою. Потім обирається пункт авторизації та користувач вводить свої дані, і у випадку, якщо дані введені вірно, система надає можливість перейти в особистий кабінет та подавати оголошення. Саме з цього етапу починається випробування КЗ.

6 РЕЗУЛЬТАТИ ВИПРОБУВАНЬ

В процесі тестування були перевірена уся функціональність комплексу задач. У наступних таблицях наведений перелік випробувань основних функціональних можливостей (таблицях 6.1 – 6.18)

6.1.Функціональне тестування

Таблиця 6.1 – Введення логіну та паролю

Мета тесту:	Перевірка функції «Вхід в частину адміністрування»
Початковий стан КЗ	Відкрита «Вхід» сторінка сайту
Вхідні данні:	Логін та пароль
Схема проведення тесту:	Ввести у поле «Логін» логін, а у поле «Пароль»
	Ввести пароль, натиснути кнопку «Вхід»
Очікуваний результат:	Відкрита сторінка «Особистий кабінет»
Стан КЗ після проведення випробувань:	Відкрита сторінка «Особистий кабінет»

Таблиця 6.2 – Перевірка заповнення усіх атрибутів

Мета тесту:	Перевірка функції «Перевірка заповнення усіх атрибутів»
Початковий стан КЗ	Відкрита «Вхід» сторінка сайту
Вхідні данні:	Логін або пароль
Схема проведення тесту:	Ввести у поле «Логін» логін, або у поле «Пароль» ввести пароль адміністратора, натиснути кнопку «Вхід»
Очікуваний результат:	Вікно повідомлення про помилку вводу. Вхід на сторінку «Особистий кабінет» не виконано «Неверное имя или пароль»
Стан КЗ після проведення випробувань:	Відкрита сторінка аутентифікації

Таблиця 6.3 – Перевірка правильності введених даних

Мета тесту:	Перевірка функції «Перевірка правильності введених даних»
Початковий стан КЗ	Відкрита «Вхід» сторінка сайту
Вхідні данні:	Логін та пароль
Схема проведення тесту:	Ввести у поле «Логін» будь-який логін, а у поле «Пароль» ввести будь-який пароль , натиснути кнопку «Вхід»
Очікуваний результат:	Вікно повідомлення про помилку вводу. Вхід на сторінку «Особистий кабінет» не виконано «Неверное имя или пароль»
Стан КЗ після проведення випробувань:	Відкрита сторінка аутентифікації

Таблиця 6.4 – Повернення на головну сторінку сайту

Мета тесту:	Перевірка функції «Повернення на головну сторінку сайту»
Початковий стан КЗ:	Відкрита сторінка аутентифікації
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку «Повернутися на головну»
Очікуваний результат:	Відкрита головна сторінка сайту
Стан КЗ після проведення випробування:	Відкрита головна сторінка сайту

Таблиця 6.5 – Додавання категорії

Мета тесту:	Перевірка функції «Додавання категорії»
Початковий стан КЗ:	Відкрита сторінка «Створити категорію»

Продовження таблиці 6.5

Вхідні дані:	Назва категорії та батьківська категорія
Схема проведення тесту:	Натиснути кнопку «Створити», ввести у полі «Назва» назву, у полі «Батьківська категорія» - обрати категорію, натиснути кнопку «Додати»
Очікуваний результат:	На сторінці «Категорія» з'явилась нова категорія
Стан КЗ після проведення випробування:	Відкрита сторінка «Категорії»

Таблиця 6.6 – Додавання фото при створенні оголошення

Мета тесту:	Перевірка функції «Додавання фото при створенні оголошення»
Початковий стан КЗ:	Відкрита сторінка «Додати оголошення»
Вхідні дані:	Файл фото, назва фото
Схема проведення тесту:	Натиснути кнопку «Добавити», заповнити усі поля на сторінці, натиснути кнопку «Добавити»

Продовження таблиці 6.6 – Додавання фото при створенні оголошення

Очікуваний результат:	Фото додано. Повідомленні про успіх додавання та фото відображається на тій самій сторінці.
Стан КЗ після проведення випробування:	Відкрита сторінка «Додавання оголошення»

Таблиця 6.7 – Перевірка заповнення усіх полів на сторінці

Мета тесту:	Перевірка функції «Перевірка заповнення усіх полів на сторінці»
Початковий стан КЗ:	Відкрита сторінка «Додавання оголошення»
Вхідні дані:	Файл фото або назва фото

Продовження таблиці 6.7

Схема проведення тесту:	Натиснути кнопку «Добавити», заповнити усі поля на сторінці, натиснути кнопку «Добавити»
Очікуваний результат:	Вікно повідомлення про помилку вводу. Додавання фото не виконано.
Стан КЗ після проведення випробування:	Відкрита сторінка «Додавання оголошення»

Таблиця.6.8 – Редагування зображення

Мета тесту:	Перевірка функції «Редагування фотографії»
Початковий стан КЗ:	Відкрита сторінка «Редагування оголошення»
Вхідні дані:	Марка, Модель, Колір, Рік, Об'єм двигуна, Тип кузова, Тип пального, Тип коробки передач, Описання, Ціна, Пошта, Номер Телефону, Назва категорії.
Схема проведення тесту:	Натиснути кнопку «Огляд». Обрати зображення. Натиснути кнопку «Редагувати».
Очікуваний результат:	Запис відредагований
Стан КЗ після проведення випробування:	Відкрита сторінка «Мій Асортимент»

Таблиця.5.9 – Перевірка функції подачі оголошення незареєстрованим користувачем.

Мета тесту:	Перевірка функції «Перевірка функції подачі оголошення незареєстрованим користувачем. »
Початковий стан КЗ:	Відкрита сторінка «Асортимент»
Вхідні дані:	

Продовження таблиці 6.9

Схема проведення тесту:	Натиснути кнопку «Створити оголошення»
Очікуваний результат:	Система просить користувача авторизуватись
Стан КЗ після проведення випробування:	Відкрита сторінка «Вхід»

Таблиця.6.10 – Перевірка унікальності імені користувача

Мета тесту:	Перевірка функції «Перевірка унікальності імені користувача»
Початковий стан КЗ:	Відкрита сторінка «Реєстрація»
Вхідні дані:	Ім'я користувача, Ім'я, Прізвище, Пошта, Номер телефону.
Схема проведення тесту:	Заповнити усі поля на сторінці, натиснути кнопку «Зареєструватись»
Очікуваний результат:	Система виводить повідомлення: «Цей логін зайнятий іншим користувачем»
Стан КЗ після проведення випробування:	Відкрита сторінка «Реєстрація»

Таблиця.6.11. – Перевірка можливості редагування усіх оголошень адміністратором.

Мета тесту:	Перевірка функції «Перевірка можливості редагування усіх оголошень адміністратором»
Початковий стан КЗ:	Відкрита сторінка «Асортимент»
Вхідні дані:	Оголошення від усіх користувачів
Схема проведення тесту:	Натиснути кнопку «Редагувати», змінити усі, або окремі поля, натиснути кнопку «Редагувати»

Продовження таблиці 6.11

Очікуваний результат:	Відкрита сторінка «Асортимент». Зміни видні як адміністратору, так і усім користувачам.
Стан КЗ після проведення випробування:	Відкрита сторінка «Асортимент»

Таблиця.6.12 – Додавання фото при редагуванні

Мета тесту:	Перевірка функції «Редагування фото»
Початковий стан КЗ:	Відкрита сторінка «Редагування оголошень»
Вхідні дані:	Файл фото, назва
Схема проведення тесту:	Натиснути кнопку «Редагувати», заповнити усі поля на сторінці, натиснути кнопку «Редагувати»
Очікуваний результат:	Додана нова фотографія
Стан КЗ після проведення випробування:	Відкрита сторінка «Мій Асортимент»


Таблиця.6.13– Видалення фото при редагуванні

Мета тесту:	Перевірка функції «Редагування фото»
Початковий стан КЗ:	Відкрита сторінка «Редагування оголошень»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку «Видалити» біля фото, яке потрібно видалити. Підтвердити бажання видалити авто натиснувши кнопку «ОК»
Очікуваний результат:	Фото видалено

Продовження таблиці 6.13

Стан КЗ після проведення випробування:	Відкрита сторінка «Редагування оголошень»
--	---

Таблиця.6.14– Перегляд оголошення

Мета тесту:	Перевірка функції «Перегляд оголошення»
Початковий стан КЗ:	Відкрита сторінка «Асортимент»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку 
Очікуваний результат:	Відкрита сторінка «Перегляд оголошення»
Стан КЗ після проведення випробування:	Відкрита сторінка «Перегляд оголошення»

6.2.Модульне тестування

Таблиця 6.15 – Додавання оголошення

Мета тесту:	Перевірка функції «Додавання оголошення»
Початковий стан КЗ:	Відкрита сторінка «Додати оголошення»
Вхідні дані:	Новина та підпис
Схема проведення тесту:	Натиснути кнопку «Добавити», ввести у полі «Оголошення» назву, у полі «Категорія» - обрати категорію, у полі «Опис» - додати опис товару, у полі «Фото» - додати зображення товару, натиснути кнопку «Додати»
Очікуваний результат:	На сторінці «Оголошення» з'явилося нове оголошення
Стан КЗ після проведення випробування:	Відкрита сторінка «Особистий кабінет»

Таблиця 6.16 – Редагування оголошення

Мета тесту:	Перевірка функції «Редагування видалення»
Початковий стан КЗ:	Відкрита сторінка «Особистий кабінет»
Вхідні дані:	


Продовження таблиці 6.16

Схема проведення тесту:	Натиснути кнопку «Редагувати» біля оголошення, яке потрібно відредагувати. У полі оголошення відредагувати оголошення, у полі інформація відредагувати інформацію. Натиснути кнопку
	«Застосувати»
Очікуваний результат:	Оголошення відредаговане
Стан КЗ після проведення випробування:	Відкрита сторінка «Особистий кабінет»

Таблиця 6.17 – Видалення оголошення

Мета тесту:	Перевірка функції «Видалення оголошення»
Початковий стан КЗ:	Відкрита сторінка «Особистий кабінет»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку «Видалити» біля оголошення, яке потрібно видалити. Підтвердити бажання видалити новину натиснувши кнопку «ОК»
Очікуваний результат:	Оголошення видалене
Стан КЗ після проведення випробування:	Відкрита сторінка «Особистий кабінет»

Таблиця 6.18 – Перегляд категорій

Мета тесту:	Перевірка функції «Перегляд категорій»
Початковий стан КЗ:	Відкрита сторінка «Категорії»
Вхідні дані:	
Схема проведення тесту:	Натиснути кнопку 
Очікуваний результат:	Відкрита сторінка «Перегляд категорій»

Продовження таблиці 6.18

Стан КЗ після проведення випробування:	Відкрита сторінка «Перегляд категорій»
---	--

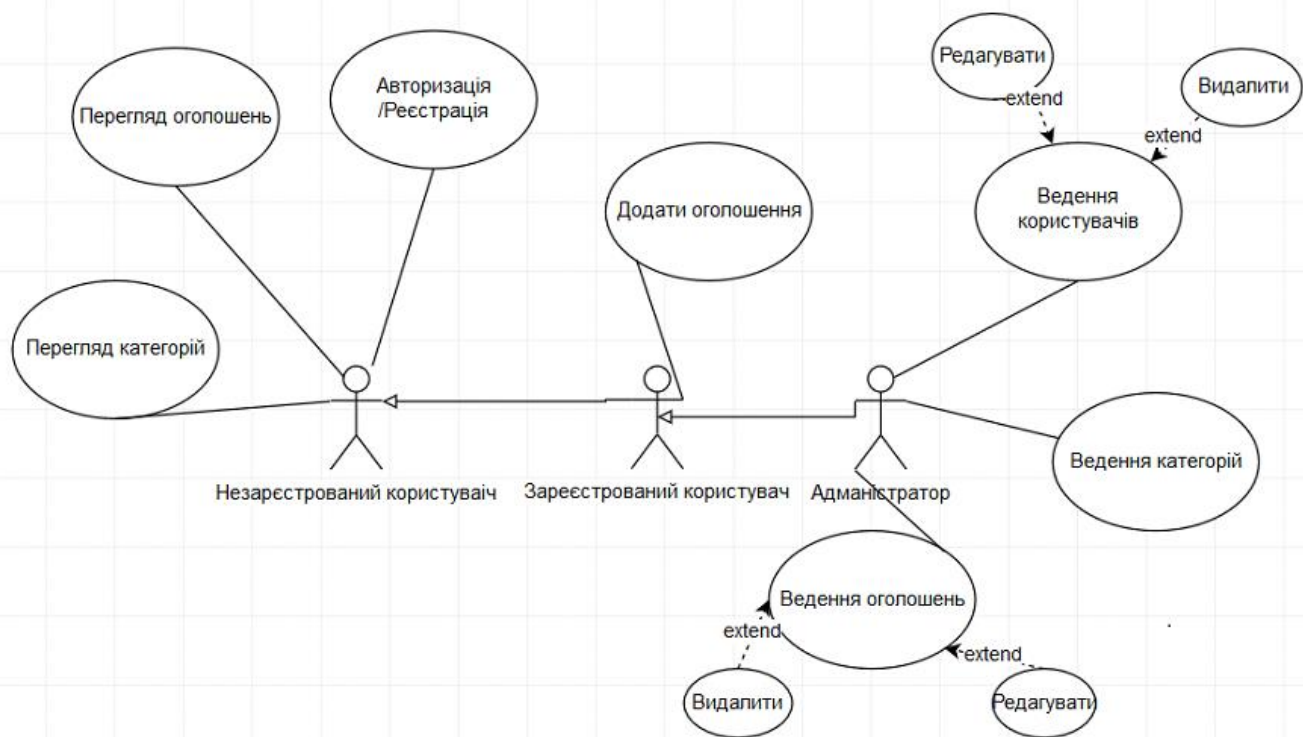
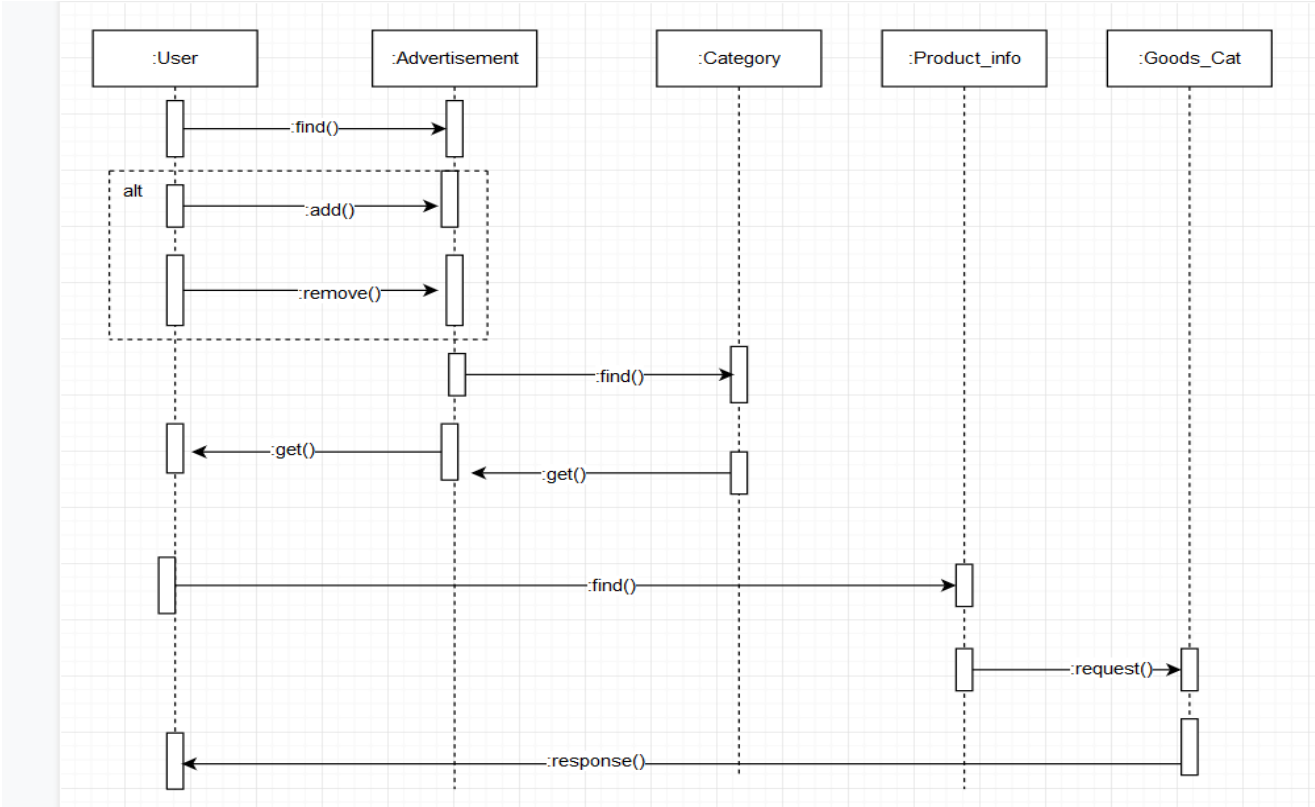
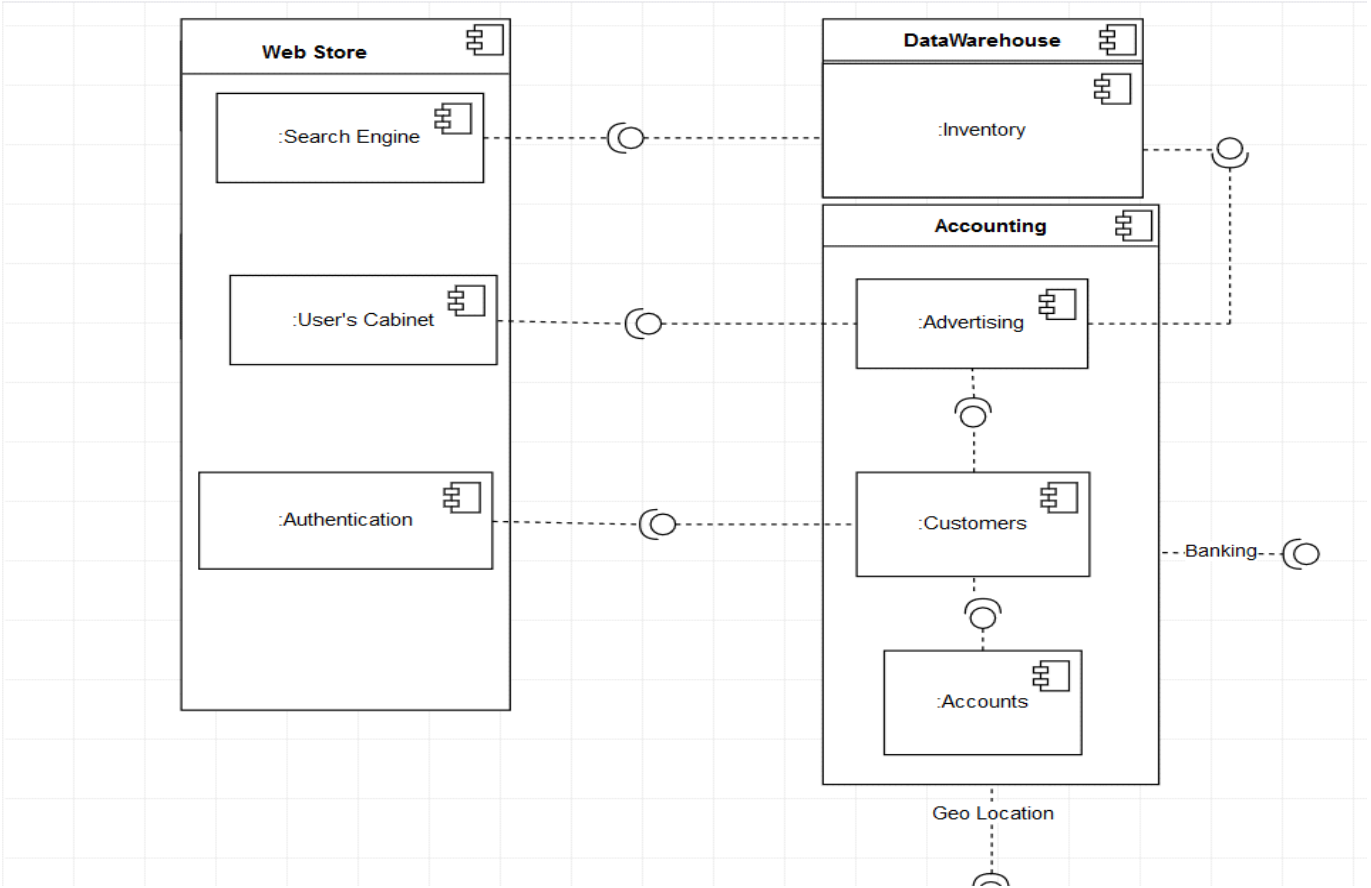


Рисунок 1 – Схема структурна варіантів використання

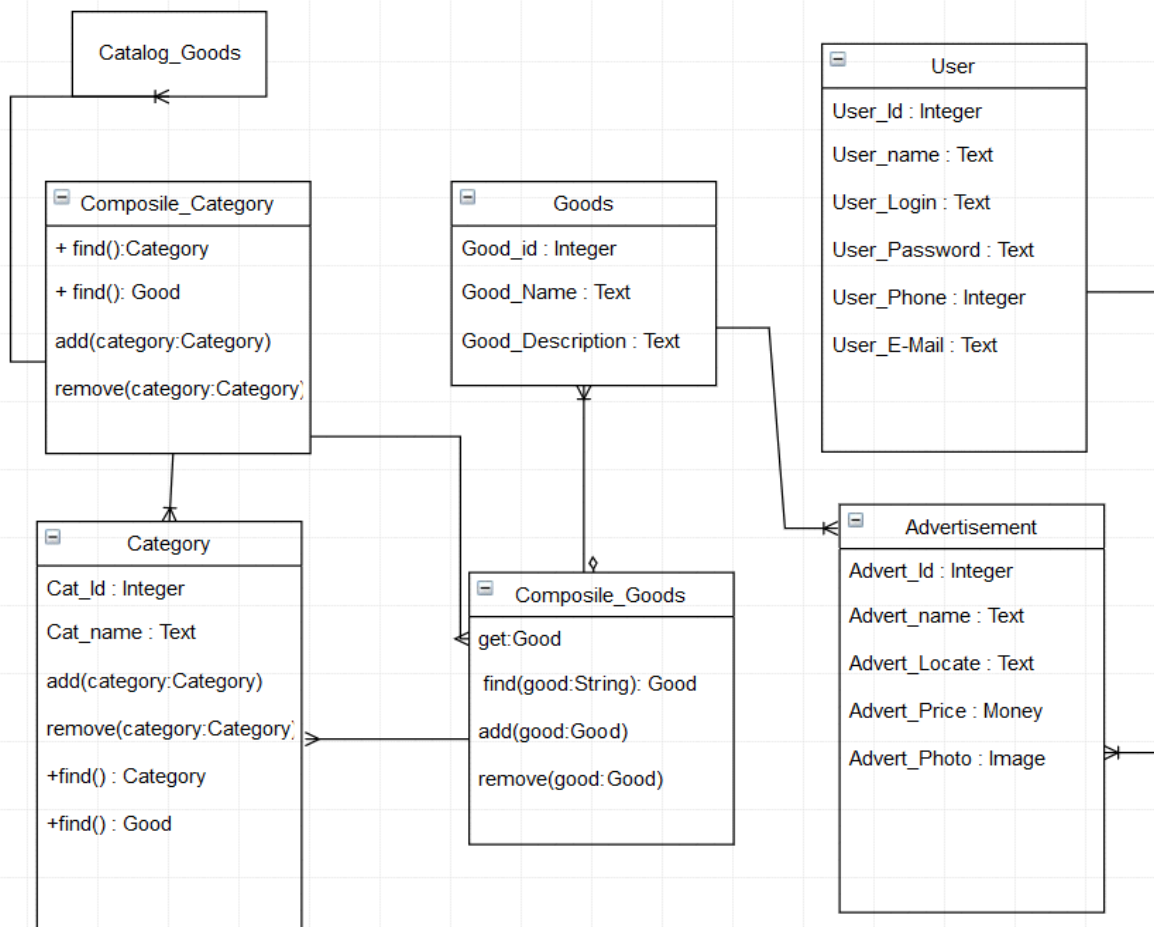
					ДП ІС-5223.1181-с.СБВВ						
					Схема структурна варіантів використання						
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Уманський В.А									
Перевірив		Цюцюра М.І.									
Т. кон.											
Н. кон.		Халус О.А.			Інформаційна система підтримки процесу подачі оголошень						
Затвердив		Уманський В.А.									
					Літера			Маса		Масштаб	
					Аркуш 1			Аркушів 1			
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52						



					ДП ІС-5223.1181-с.ССП				
					Схема структурна послідовності				
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Уманський В.А							
Перевірив		Цюцюра М.І.			Інформаційна система підтримки процесу подачі оголошень				
Т. кон.									
Н. кон.		Халус О.А.							
Затвердив		Уманський В.А.							
					Літера				
					Маса				
					Масштаб				
					Аркуш 1				
					Аркушів 1				
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52				



					ДП ІС-5223.1181-с.ССК								
					Схема структурна компонентів				Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата									
Розробив		Уманський В.А											
Перевірив		Цюцюра М.І.											
Т. кон.									Аркуш 1		Аркушів 1		
Н. кон.		Халус О.А.			Інформаційна система підтримки процесу подачі оголошень				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52				
Затвердив		Уманський В.А.											



ДП ІС-5223.1181-с.ССК

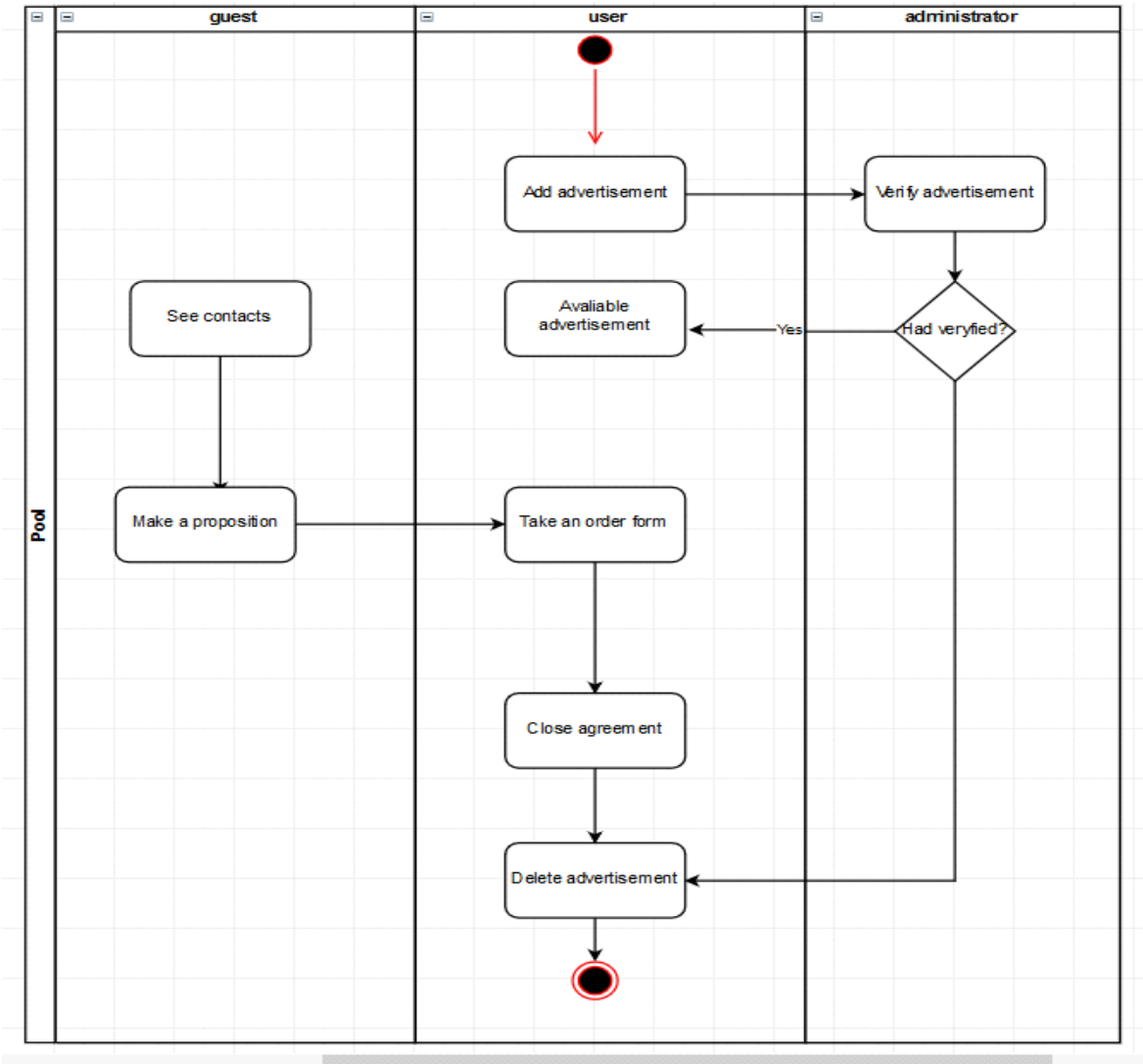
Схема структурна класів

Інформаційна система підтримки
процесу подачі оголошень

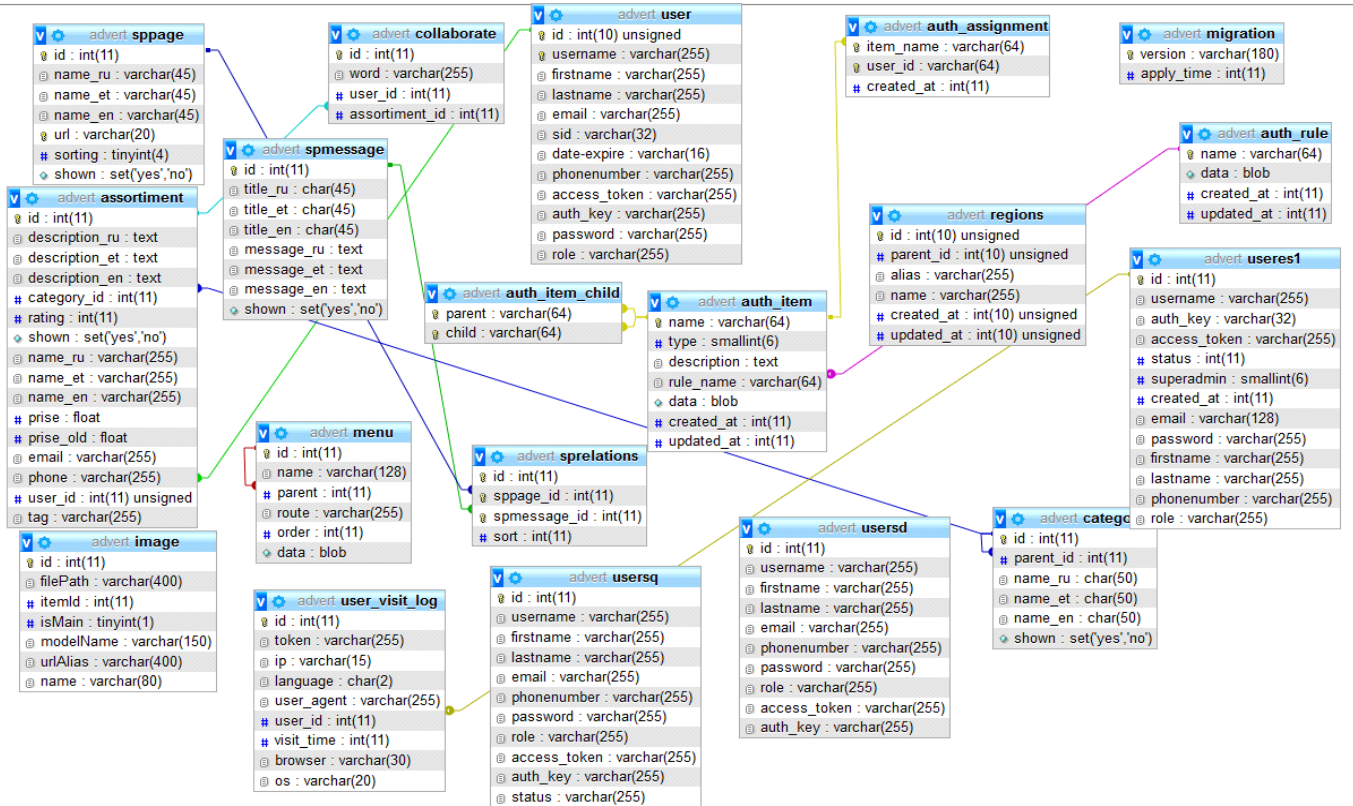
Літера	Маса	Масштаб
Аркуш 1		Аркушів 1

КПІ ім. Ігоря Сікорського
кафедра АСОІУ гр. ІС-52

Зм.	Арк.	№ документа	Підпис	Дата
Розробив	Уманський В.А.			
Перевірив	Цюцюра М.І.			
Т. кон.				
Н. кон.	Халус О.А.			
Затвердив	Уманський В.А.			



					ДП ІС-5223.1181-с.ССД				
					Схема структурна діяльності				
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Уманський В.А.							
Перевірив		Цюцюра М.І.			Інформаційна система підтримки процесу подачі оголошень				
Т. кон.									
Н. кон.		Халус О.А.							
Затвердив		Уманський В.А.							
					Літера				
					Маса				
					Масштаб				
					Аркуш 1				
					Аркушів 1				
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52				



ДП ІС-5223.1181-с.СБД

					ДП ІС-5223.1181-с.СБД			
					Схема бази даних			
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Уманський В.А.						
Перевірив		Цюцюра М.І.						
Т. кон.					Інформаційна система підтримки процесу подачі оголошень			
Н. кон.		Халус О.А.						
Затвердив		Уманський В.А.						
					Літера Маса Масштаб			
					Аркуш 1 Аркушів 1			
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-52			